

Mobile Mixed Reality Platform



Vom Fachbereich Informatik
der technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieur (Dr.-Ing.)

von

Dipl.-Inform. Jens-Martin Weidenhausen

geb. in Frankfurt am Main

Referenten der Arbeit:

Prof. Dr.-Ing. Dr. h. c. José L. Encarnação
Prof. Dr. Stefan Müller

Tag der Einreichung:

11.10.2006

Tag der mündlichen Prüfung:

12.12.2006

Darmstädter Dissertation, 2006

D17

Inhaltsverzeichnis

1	EINLEITUNG	7
1.1	Motivation	7
1.2	Problemstellung und Zielsetzung	8
1.3	Gliederung	9
1.4	Zusammenfassung der wichtigsten Ergebnisse	10
2	GRUNDLAGEN	12
2.1	Augmented Reality	12
2.2	AR Basisfunktionalitäten	13
2.2.1	Tracking und Registrierung	13
2.2.1.1	Tracking in VR Umgebungen	14
2.2.1.2	Tracking für Outdoor Anwendungen	14
2.2.1.3	Visionbasierte Tracking-Verfahren	14
2.2.1.4	Relative Tracking-Verfahren	15
2.2.1.5	Hybride Tracking-Verfahren	15
2.3	Präsentation	16
2.3.1	Displays	16
2.3.2	Darstellungstechniken	18
2.3.3	Interaktion	19
2.3.3.1	Interaktionsgeräte in immersiven VR Umgebungen	20
2.3.3.2	Interaktionsgeräte für mobile Anwendungen	21
2.4	Augmented Reality basierte Assistenzfunktionen	23
2.5	Benutzerschnittstellen	25
2.6	Zusammenfassung	29
3	ANFORDERUNGEN AN EIN MOBILES AR SYSTEM	31

3.1	Anwendungsgebiete	31
3.1.1	Produktentwicklung	32
3.1.2	Fertigung	33
3.1.3	Service und Wartung	36
3.1.4	Kulturelles Erbe	37
3.2	Anwendergruppen	38
3.2.1	Endanwender	39
3.2.2	Applikationsentwickler	39
3.2.3	Systementwickler	40
3.3	Analyseergebnisse	40
3.3.1	Arbeitsmodi	40
3.3.2	Navigationsunterstützung	42
3.3.3	Nahbereichsnavigation	43
3.3.4	Kontextvisualisierung	43
3.3.5	Informationspräsentation	44
3.3.6	Informationsfilterung	44
3.3.7	Interaktion	44
3.3.8	Kooperation	45
3.3.9	Konfiguration des Systems	45
3.3.10	Erstellung von Inhalten	45
3.4	Nichtfunktionale Anforderungen	45
3.4.1	Qualität des Displays	46
3.4.2	Performanz	46
3.4.3	Gewicht und Größe	47
3.4.4	Betriebsdauer / Stromversorgung	47
3.4.5	Sonstige Anforderungen	47
3.5	Bestehende Systeme	47
3.5.1	ARToolkit	48
3.5.2	Studierstube	49
3.5.3	DWARF	50
3.5.4	Tinmith-Evo5	51
3.6	Zusammenfassung	52
4	ARCHITEKTURKONZEPT FÜR EINE MOBILE AR PLATTFORM	53
4.1	Framework	53
4.2	System-Architektur	56
4.2.1	Übersicht	56

4.2.2	Framework Basisfunktionalität	57
4.2.3	Komponenten	58
4.2.3.1	Methoden	58
4.2.3.2	Attribute	58
4.2.3.3	Events	60
4.2.4	Basisdienste	60
4.2.5	Erweitertes Weltmodell	60
4.2.5.1	Informationsraum	61
4.2.5.2	Informationsobjekt	62
4.2.6	Informationsfilterung	66
4.3	Geräteschnittstelle	69
4.3.1	Bestehende Systeme	70
4.3.1.1	OpenTracker	70
4.3.1.2	IDEAL	70
4.3.2	Hybride Tracking-Verfahren	72
4.3.3	Tracking-Framework	73
4.3.3.1	Framework-Architektur	75
4.3.3.2	Komponenten	76
4.3.3.3	Multithreading	77
4.3.3.4	Immediate-Methoden	77
4.3.3.5	Integration in den AR-Browser	78
4.4	Präsentation	79
4.4.1	Darstellungsmodi	80
4.4.2	Visueller Realismus	83
4.4.3	Anbindung Device-Interface - Präsentation	84
4.5	Einbettung von AR Inhalten in Applikationen	87
4.5.1	AR-Viewer als Applikation	87
4.5.2	Einbettung von AR in eine Applikation – AR-Browser	88
4.6	Zusammenfassung	91
5	INTERAKTIONSKONZEPT FÜR MOBILE AR ANWENDUNGEN	93
5.1	Grundlagen der Interaktion in virtuellen Umgebungen	94
5.2	Interaktionsaufgaben in mobilen AR Anwendungen	94
5.3	Interaktionstechniken in mobilen AR Anwendungen	96
5.4	Universelles Interaktionsgerät für mobile AR Anwendungen	101
5.4.1	Anforderungen	101

5.4.2	Konzept	102
5.5	GUI Darstellung in Augmented Reality	112
5.5.1	GUI-Abstraktion	113
5.5.2	GUI-Manager	116
5.5.3	GUI Objekte	116
5.6	Zusammenfassung	120
6	ENGINEERING MOBILER AR ANWENDUNGEN	121
6.1	Engineeringaufgaben	121
6.1.1	Einrichten und Kalibrieren von Tracking-Systemen	121
6.1.2	Kalibrierung von See-Through-Displays	122
6.1.3	Modellierung des virtuellen Informationsraums	122
6.2	Template basierte Erstellung von Serviceanweisungen	123
6.2.1	Konzept	124
6.2.2	Realisierung	127
6.2.3	Erweitertes Template Konzept	128
6.2.4	Online Authoring	130
6.3	Zusammenfassung	131
7	ZUSAMMENFASSUNG UND AUSBLICK	132
7.1	Zusammenfassung	132
7.2	Ausblick auf weiterführende Arbeiten	134
	LITERATURVERZEICHNIS	135
	ABBILDUNGSVERZEICHNIS	143
	EIGENE VERÖFFENTLICHUNGEN	147

1 Einleitung

1.1 Motivation

Augmented Reality (AR) ist eine neuartige Mensch Computer Schnittstelle, die es erlaubt, einem Benutzer kontextbezogene Informationen lagerichtig in sein Sichtfeld einzublenden. Im Gegensatz zu immersiven virtuellen Umgebungen, bei denen der Benutzer sich in eine spezielle Umgebung begibt, um in die virtuelle Welt einzutauchen, liegt das Potential von Augmented Reality vor allem darin, dass der Anwender in seiner gewohnten, realen Umgebung zusätzliche Informationen erhält, die in direktem räumlichen Bezug zu den Objekten der realen Umgebung stehen. Beispiele für Anwendungen sind die AR basierte Unterstützung von Servicetätigkeiten an komplexen technischen Anlagen oder die Präsentation virtueller Rekonstruktionen zerstörter antiker Bauwerke an deren ursprünglichen, realen Standorten.

Zur lagerichtigen Einblendung der Informationen in das Blickfeld des Anwenders ist die schnelle und präzise Bestimmung von Position und Blickrichtung des Anwenders notwendig. Ebenso wichtig zur vollen Ausschöpfung des Potentials von AR ist aber auch die Unterstützung der Mobilität des Benutzers. Die Technologie soll dort Verfügbar sein, wo der Benutzer ihre Unterstützung benötigt. Durch die zunehmende Verbreitung von leistungsfähigen Handheld-Rechnern, Kamera-Handys und PDAs sowie von verschiedenen drahtlosen Informationsnetzen wie WLAN und UMTS ist hierfür mittelfristig eine hardwareseitige Basis für den breiten Einsatz von Augmented Reality gegeben.

Eine Schlüsselrolle bei der Realisierung von Augmented Reality Anwendungen kommt dem Tracking des Benutzers, der schnellen und präzisen Erfassung seiner Position und seiner Blickrichtung, zu. Dies spiegelt sich auch in der Gewichtung der Forschungsarbeiten auf dem Gebiet der Augmented Reality wieder. Hier beschäftigt sich die Mehrzahl der Arbeiten vor allem mit kamerabasierten Tracking-Verfahren und deren Kombination mit anderen Tracking-Technologien. Trotz der herausragenden Stellung der Lösung des Tracking-Problems darf aber nicht außer acht gelassen werden, dass die Realisierung mobiler AR Anwendungen über das Tracking hinaus die Lösung weiterer Probleme erfordert.

Die Vielzahl unterschiedlicher Anwendungen, die potentiell mit Augmented Reality realisiert werden können, erfordern eine adäquate Softwareinfrastruktur, die es erlaubt diese Anwendungen effizient umzusetzen und dabei die bestmögliche Anpassung an die jeweiligen Bedingungen und Anforderungen zu erreichen.

Die Akzeptanz der potentiellen Benutzer zukünftiger AR Anwendungen lässt sich nur durch eine ganzheitliche Herangehensweise an die neue Technologie erreichen. Dabei sind sowohl technologische, benutzerabhängige als auch soziale Aspekte zu berücksichtigen. Eine wichtige Erkenntnis aus

den Anfangszeiten der Erforschung von Virtuellen Realität (VR) ist, dass vor allem die Usability solcher Systeme ein entscheidendes Kriterium für deren Akzeptanz darstellt. Das typische Bild eines VR Anwenders aus dieser Zeit zeigt diesen mit einem geschlossenen Head Mounted Display (HMD), wie er mit einem Datenhandschuh mit der virtuellen Umgebung interagiert. Eine Gerätekonfiguration, die heute so gut wie keine Bedeutung mehr besitzt, aber das Bild der Technologie VR in der Öffentlichkeit nachhaltig geprägt hat. Erst die vereinfachte Bedienung und die Verwendung den Aufgaben entsprechender Display-Systeme eröffnete auch Benutzergruppen, die keine VR Spezialisten sind, die Möglichkeit mit solchen Systemen zu arbeiten.

Bereits mit verfügbaren Tracking- und Display-Systemen in Verbindung mit lassen sich prototypische mobile Augmented Reality Systeme realisieren, die es erlauben auch andere Aspekte dieser Technologie anzugehen. Dazu gehört die kontextbasierte Aufbereitung und Bereitstellung von Daten ebenso, wie die Interaktion mit Umgebungen, die sowohl aus realen als auch aus virtuellen Bestandteilen bestehen.

Ein typisches Gebiet für den Einsatz von Augmented Reality ist die Unterstützung bei der Durchführung einer realen Aufgabe. Um aber eine wirkliche Hilfe darzustellen, muss das System ein hohes Maß an Benutzerfreundlichkeit gewährleisten, damit es nicht von der eigentlichen Aufgabe ablenkt oder deren Durchführung sogar erschwert. Augmented Reality als Ganzes sollte dabei als Mensch Computer Schnittstelle verstanden werden, wobei alle Aspekte der Realisierung zu deren Brauchbarkeit beitragen und deshalb auch mit der gleichen Sorgfalt angegangen werden müssen. Nur so kann Augmented Reality als Mensch Maschine Schnittstelle in Verbindung mit mobiler Hardware das in ihr enthaltene Potential, das Alltagsleben zu durchdringen, ausschöpfen.

1.2 Problemstellung und Zielsetzung

Die Realisierung mobiler Augmented Reality Anwendungen erfordert neben einem präzisen und schnellen Tracking auch die Bereitstellung der relevanten, kontextabhängigen Informationen sowie geeigneter Präsentationstechniken für virtuelle Objekte in realen Umgebungen vor allem auch angemessene Interaktionstechniken für die Bedienung solcher Systeme. Der Benutzer eines Augmented Reality Systems interagiert gleichzeitig mit der realen Umgebung und den virtuellen Objekten, die ihm in einem Display eingeblendet werden.

Das AR Basis-System soll den Rahmen für die Erstellung unterschiedlicher mobiler Anwendungen liefern. Es stellt Tracking-, Interaktions- und Präsentationstechniken bereit, die den jeweiligen Anforderungen einer Anwendung angepasst werden können. Obwohl diese Anforderungen denen bei der Realisierung von immersiven VR Anwendungen ähneln, ergeben sich vor allem durch die geforderte Mobilität des Systems und die damit verbundenen Bedingungen, Anforderungen, die sich mit bestehenden VR Systemarchitekturen gar nicht oder nur unangemessen erfüllen lassen. Auch die bereitzustellenden Interaktions- und Präsentationstechniken müssen dem mobilen AR Umfeld speziell angepasst sein. Obwohl auch hier Parallelen zur Interaktion in VR Anwendungen bestehen, erfordert die Realisierung von Systemen zur Unterstützung realer Aufgaben durch ein mobiles AR System neuartige Bedienkonzepte.

Ziel dieser Arbeit ist die Bereitstellung einer Plattform für mobile Augmented Reality Anwendungen, bestehend aus einer tragfähigen System-Architektur und den darin realisierten Basisfunktionalitäten. Besonderes Augenmerk soll dabei auf einen Satz von Interaktions- und Präsentationstechniken gelegt werden, die den speziellen Anforderungen durch mobile AR Anwendungen entsprechen. Da der Plattformgedanke nicht an Software-Technologien gebunden ist, werden auch Eingabegeräte betrachtet und ein Konzept für ein universelles Eingabegerät vorgestellt, das ebenfalls im Rahmen der Plattform den Entwicklern und Anwendern zur Verfügung gestellt werden sollte, um das volle Potential der Plattform, und somit der Technologie Mobile Augmented Reality ausschöpfen zu können.

1.3 Gliederung

Im folgenden Kapitel werden die Grundlagen für die Realisierung von Augmented Reality Systemen betrachtet. Dabei werden Schwerpunktmäßig die drei Basistechnologien Tracking, Präsentation und Interaktion beleuchtet, die zur Erstellung eines Augmented Reality Basissystems unverzichtbar sind.

Ein Überblick über unterschiedliche bestehende Anwendungen für Augmented Reality basierte Assistentenfunktionalitäten zeigt den Rahmen der in dieser Arbeit betrachteten Anwendungsdomäne auf. Abgeschlossen wird dieses Kapitel durch eine Übersicht über die allgemeinen Grundlagen der Gestaltung von Benutzerschnittstellen.

Kapitel 3 befasst sich mit den Anforderungen an ein mobiles Augmented Reality System. Hierbei wird unterschieden zwischen den Anforderungen aus den verschiedenen Anwendungsgebieten und den Anforderungen von verschiedenen Anwendergruppen, die mit dem System in Kontakt kommen.

Die Analyse der hier gewonnenen Anforderungen führt zu diversen Funktionalitäten, die ein mobiles Augmented Reality System bereitstellen muss, die über die reinen Basisfunktionalitäten hinausgehen.

Neben den rein funktionalen Anforderungen ergeben sich aus der Anforderungserhebung auch Anforderungen, die nicht durch die Realisierung des Softwaresystems abgedeckt werden können. Diese Anforderungen fließen aber wiederum in abgeleiteter Form in die Software und auch direkt in die Konzeption eines Interaktionsgerätes ein. Eine Betrachtung der relevanten, bestehenden Augmented Reality Systeme unter den Aspekten der hier erhobenen Anforderungen schließt das Kapitel ab.

Basierend auf den in Kapitel 3 erhobenen Anforderungen wird in Kapitel 4 eine Architektur für eine mobile Augmented Reality Plattform konzipiert. Dabei wird einleitend auf die spezielle Herausforderung der Realisierung eines Rahmensystems eingegangen und ein Vorgehen für den Umgang mit der daraus entstehenden Komplexität der Aufgabe dargestellt. Anschließend wird die eigentliche Systemarchitektur vorgestellt und auf einige spezielle, neuartige Lösungen näher eingegangen. Hervorzuheben sind dabei Schlüsseltechnologien, wie Informationsobjekte in Verbindung mit dem erweiterten Weltmodell und Funktionalitäten für die effiziente Filterung von Informationen zur Verringerung der darzustellenden Informationsmenge.

Kapitel 5 befasst sich eingehend mit neuartigen Bedienkonzepten für mobile Augmented Reality Anwendungen. Zu Beginn werden hier Interaktionstechniken aus immersiven VR Anwendungen auf die Tauglichkeit ihrer Anwendung in mobilen AR Anwendungen betrachtet. Resultierend aus den hierbei beobachteten Unzulänglichkeiten wird ein Konzept für ein universelles Interaktionsgerät zur Bedie-

nung mobiler AR Anwendungen erstellt und prototypisch umgesetzt. Neben der Hardwarerealisierung aus verfügbaren Standardkomponenten werden auch Interaktionstechniken entwickelt, die speziell auf die Funktionalitäten des Interaktionsgerätes abgestimmt sind. Darüber hinaus werden weitere Interaktions- und Präsentationstechniken vorgestellt, die speziell auf die Bedürfnisse mobiler Anwendungen in Verbindung mit verfügbaren Sichtgeräten abgestimmt sind.

In Kapitel 6 wird speziell die Vorbereitung und Erstellung von mobilen AR Anwendungen betrachtet. Dabei werden einleitend die unterschiedlichen Engineeringschritte betrachtet, die notwendig sind, bis eine Anwendung ausgeführt werden kann. Anschließend wird ein Verfahren vorgestellt, das den Aufwand der Erstellung animierter, virtueller Arbeitsanweisungen erheblich reduziert.

Abschließend werden in Kapitel 7 die wichtigsten Ergebnisse zusammengefasst und ein Ausblick auf weitere Arbeiten gegeben.

1.4 Zusammenfassung der wichtigsten Ergebnisse

Im Rahmen dieser Arbeit wurde anhand allgemeingültiger Anforderungen an mobile Augmented Reality Systeme ein Architekturkonzept entwickelt und prototypisch in Form des AR-Browsers realisiert. Der AR-Browser ist ein komponentenbasiertes Augmented Reality Basissystem, auf dessen Basis verschiedenste mobile AR Anwendungen realisiert werden können.

Die Architektur erlaubt die einfache Erweiterung um zusätzliche Funktionalitäten und das Ersetzen von Basisfunktionalitäten durch alternative Realisierungen. Die in einem vorhergehenden Schritt ermittelten Anforderungen an ein mobiles AR System spiegeln sich sowohl in der Softwarearchitektur, als auch in zusätzlichen Komponenten wieder, die spezifische Funktionalitäten für mobile AR Anwendungen bereitstellen, wie das erweiterte Weltmodell und Mechanismen für die Informationsfilterung.

Ein zusätzliches Tracking-Framework stellt die Funktionalitäten bereit, die zur Erstellung komplexer, kombinierter Tracking-Verfahren notwendig sind. Das Tracking-Framework erlaubt den Aufbau von Tracking-Graphen aus verschiedenen Komponenten, die entweder Sensoren, Bildquellen, bildbasierte Tracking-Verfahren oder die Sensor-Fusion beinhalten.

Besonderes Augenmerk wurde zusätzlich auf die Einbettung von AR-Inhalten in bestehende Applikationen gelegt. Zumindest für Windows basierte Anwendungen konnte dies sowohl für webbasierte, als auch für .NET und Applikationen, die COM unterstützen realisiert werden. Die AR-Präsentation kann dabei über eine COM bzw. Scripting-Schnittstelle in weitem Umfang gesteuert werden.

Neben der Betrachtung der Laufzeitseite wurde im Rahmen einer Untersuchung der kompletten Prozesskette auch die Engineeringseite genauer beleuchtet. Da hier vor allem durch das Erstellen der virtuellen Inhalte enorme Aufwände entstehen, die einem zeitlichen Gewinn durch den Einsatz von AR gegenübergestellt werden müssen. Dies bedingt, dass diesem vorbereitende Schritt besonderes Augenmerk geschenkt werden muss. Es wurde hierfür ein Verfahren entwickelt, das die Erstellung animierter, virtueller Arbeitsanweisungen enorm beschleunigen kann. Dieses Template basierte Authoringverfahren erlaubt es dem technischen Redakteur, der die Anweisungen erstellt, aus einem Satz vorbereite-

ter, parametrierbarer Anweisungen und den Werkstücken auf einfache, intuitive Weise neue Anweisungen zu komponieren.

Ein weiterer Schwerpunkt der vorliegenden Arbeit stellt die Verbesserung der Bedienbarkeit mobiler AR Systeme dar. Hier ist besonders die Konzeption eines neuen, universellen Interaktionsgerätes für mobile AR Anwendungen und die damit verbundenen, extra auf dieses Gerät abgestimmten neuen Interaktionstechniken, zu erwähnen. Zusammen mit einem System für die einfache Erstellung von GUI Elementen, die sich nahtlos in die virtuelle Umgebung einfügen, lässt sich ein erheblicher Gewinn an Bedienbarkeit erzielen. Obwohl die Bedienelemente in der 3D Umgebung an Objekte der realen Umgebung angehaftet werden können, lassen sie sich doch am effizientesten über die 2D Eingabemöglichkeit des Interaktionsgerätes bedienen. Gerade die Kombination von 2D und 3D Bedienoberflächen und einem Interaktionsgerät, das sowohl 3D Selektion als auch 2D Interaktion über eine Stift und Tablett Metapher unterstützt, ist hierbei der Schlüssel zum Erfolg.

2 Grundlagen

2.1 Augmented Reality

Der Begriff Augmented Reality (AR) umschreibt eine neuartige Mensch Computer Schnittstelle, bei der die reale Umgebung durch zusätzliche, kontextabhängige Informationen angereichert wird. Obwohl diese Informationen grundsätzlich beliebiger Art sein können, konzentrieren sich die Forschungsarbeiten auf dem Gebiet AR schwerpunktmäßig auf die Präsentation visueller Informationen. Auch in dieser Arbeit wird AR hauptsächlich unter dem Aspekt der visuellen Erweiterung der Umgebung des Anwenders betrachtet.

Ein Benutzer einer AR Anwendung erhält in seinem Sichtfeld zusätzliche, virtuelle Informationen präsentiert, die in fester räumlicher Beziehung mit Objekten der realen Welt stehen. Während in rein virtuellen Umgebungen, der Virtual Reality (VR), die reale Umgebung komplett durch die virtuelle Welt ersetzt wird, handelt es sich bei AR lediglich um Ergänzungen zur realen Umgebung, im Deutschen wird Augmented Reality folglich auch als *Erweiterte Realität* bezeichnet. Virtuelle und reale Welt, die gleichzeitig wahrgenommen werden können, bilden für den Anwender eine Einheit.

Ronald Azuma, einer der Pioniere auf dem Gebiet der erweiterten Realitäten, charakterisiert AR folgendermaßen [AZUMA97]:

- Kombination von Real und Virtuell
- Interaktiv in Echtzeit
- Registriert in 3D

Die aufgezählten Kriterien dienen dabei vor allem der Abgrenzung gegenüber anderen Bereichen der 3D Computer-Grafik. AR unterscheidet sich von VR somit vor allem durch die Kombination von realer und virtueller Umgebung. Aber auch die feste Kombination von computergenerierten Inhalten und realem Bild oder Bewegtbild, wie sie vielfach in Spezialeffekten für Kinofilme eingesetzt wird, ist ausgeklammert, da hier keine Echtzeitinteraktion mit den virtuellen Informationen möglich ist. Der Dritte Punkt, die Registrierung in 3D, beschreibt die Art wie virtuelle Informationen mit der realen Umgebung kombiniert werden. Registrierung in 3D bedeutet dabei, dass virtuelle Objekte eine feste Position und Orientierung in Relation zur realen Welt besitzen. Unabhängig von der Position und Blickrichtung des Anwenders erscheinen virtuelle Objekte immer an der gleichen Position der realen Umgebung, so dass reale Welt und virtuelle Objekte als Einheit wahrgenommen werden können.

Nach dieser Charakterisierung ist Augmented Reality folglich nicht von der Verwendung spezifischer Hardwaretechnologien, wie beispielsweise Datenbrillen, so genannten Head Mounted Displays (HMD), abhängig. Auch stationäre Monitore können als Sichtgerät verwendet werden. Allerdings ist

es somit nicht immer möglich, die virtuellen Informationen direkt ins Blickfeld des Anwenders einzublenden. Der Eindruck einer Koexistenz von realer Welt und virtuellen Objekten kann nur durch Displays erzielt werden, die direkt vor den Augen des Anwenders getragen werden.

Im Gegensatz zu immersiven VR Installationen, in die sich ein Anwender hinein begeben muss, um in die virtuelle Welt eintauchen zu können und die somit den Kontakt zur realen Umgebung unterbrechen, kann der Anwender in AR weiterhin die reale Umgebung wahrnehmen. Augmented Reality erscheint somit prädestiniert, einen Anwender bei der Durchführung von Aufgaben in der realen Welt zu unterstützen.

2.2 AR Basisfunktionalitäten

Bei der Identifikation der Komponenten und der damit verbundenen Funktionalitäten kann Azumas Charakterisierung von Augmented Reality [AZUMA97] bereits ein Stück weiterhelfen. Daraus lassen sich drei Kernfunktionalitäten ableiten, die in jedem Augmented Reality System vorhanden sein müssen, dass die Charakterisierung erfüllen soll:

- Registriert in 3D → Tracking
- Kombination von real und virtuell → Präsentation
- Interaktiv in Echtzeit → Interaktion

Diese Funktionalitäten erlauben bereits die Erstellung eines Systems, das die charakteristischen Eigenschaften für Augmented Reality erfüllt. Für die Realisierung realer, auf AR Technologien basierender Anwendungen ist dies aber bei weitem noch nicht ausreichend. Diese Funktionen können jedoch, da sie in jeder AR Anwendung benötigt werden, als Kern eines jeden AR Systems betrachtet werden.



Abbildung 1: Augmented Reality Basisfunktionalitäten (Komponentensicht)

2.2.1 Tracking und Registrierung

Die lagerichtige Überlagerung der realen Umgebung mit virtuellen Informationen erfordert die präzise Erfassung von Position und Blickrichtung des Anwenders. Diese Verfolgung des Benutzers wird im Weiteren als Tracking bezeichnet. Das Ergebnis des Trackings ist die Transformation zwischen dem lokalen Koordinatensystem des Anwenders und dem Referenz-Koordinatensystem des Tracking-Systems. Abhängig vom Verhältnis zwischen dem verwendeten globalen Koordinatensystem und dem Tracking-Koordinatensystem ist noch eine weitere Transformation in das Referenz-Koordinatensystem notwendig.

2.2.1.1 Tracking in VR Umgebungen

Speziell für die Verwendung im Kontext der Virtuellen Realitäten wurden verschiedene Technologien entwickelt, die dort ebenfalls zur Verfolgung des Benutzers oder von Objekten eingesetzt werden. Hierbei handelt es sich fast ausschließlich um stationäre elektromagnetische, ultraschall- oder kamera-basierte Systeme. Gemein haben diese Lösungen, dass sie eine erhebliche Instrumentierung der Umgebung erfordern und meist nur einen kleinen Arbeitsbereich abdecken können. Gerade in mobilen AR Anwendungen ist ein großer Arbeitsbereich des Tracking-Systems allerdings ein KO Kriterium für dessen Einsatz. So lässt sich zwar der Arbeitsbereich einiger Systeme durch zusätzliche Instrumentierung der Umgebung erweitern, der damit verbundene Aufwand ist allerdings so immens, dass dies für reale Anwendungen unrealistisch ist. Vorteile dieser Tracking Systeme sind die hohen Update-Raten und die geringe Latenz sowie das Bereitstellen von Position und Orientierung, also der vollen sechs Freiheitsgrade.

2.2.1.2 Tracking für Outdoor Anwendungen

Dem gegenüber existieren für die Gewinnung von Position und Orientierung in Outdoor Szenarien vorwiegend Verfahren, die maximal drei Freiheitsgrade, also entweder Orientierung oder Position liefern können. Beispiele für solche Verfahren sind das Satelliten gestützte Navigationssystem GPS, das die Position auf der Erdoberfläche ermittelt und ein elektronischer Kompass, der die 2D Orientierung ermittelt. Beide Verfahren sind recht ungenau und anfällig gegen äußere Störeinflüsse. GPS arbeitet zusätzlich noch mit einer recht geringen Update-Rate.

2.2.1.3 Visionbasierte Tracking-Verfahren

Bei visionbasierten Tracking-Verfahren, die die Anforderungen von mobilen Augmented Reality erfüllen, führt der Benutzer die Kamera bei sich. Da die Kamera am zu verfolgenden Objekt selbst befestigt ist und die Umgebung beobachtet, spricht man auch von *inside-out* Tracking. Im Gegensatz zu typischen VR Tracking-Systemen, bei denen Sensoren in der Umgebung ein Objekt verfolgen, die auch als *outside-in* Tracking bezeichnet werden. Der Anwender führt die aktiven Komponenten des Tracking-Systems mit sich. Dabei handelt es sich um eine oder mehrere Kameras, so dass ein solches Tracking-System wenig zusätzliches Gewicht verursacht und auch gegen elektromagnetische Strahlung relativ unempfindlich ist.

Die Kamera eines solchen inside-out Verfahrens liefert Videobilder der Umgebung an die Bildverarbeitungsalgorithmen, deren Aufgabe darin besteht, Merkmale aus den Bildern zu extrahieren. In einem anschließenden Schritt wird versucht Korrespondenzen zwischen diesen 2D Features und bekannten 3D Punkten herzustellen, so dass daraus wiederum die Pose der Kamera, bestehend aus Position und Orientierung, geschätzt werden kann. Vor allem die Bildverarbeitungsalgorithmen verursachen dabei enormen Rechenaufwand, so dass die Verfahren erhebliche Anforderungen an die Prozessorleistung und in Folge auch an die Stromversorgung stellen.

Visionbasierte Tracking-Verfahren lassen sich in markerbasierte und solche, die mit natürlichen Merkmalen der Umgebung arbeiten, unterteilen. Die markerbasierten Verfahren, wie sie im ArToolkit

[KATO99] oder in ARVIKA verwendet werden, setzen das instrumentieren der Umgebung mit künstlichen, passiven Markern voraus. Diese Marker können besonders schnell und robust aus dem Videobild extrahiert werden. Featurebasierte Tracking-Verfahren [SIMON00] [FER01] [GENC02] [BEHRING02] verzichten auf das Instrumentieren der Umgebung, verursachen aber einen wesentlich höheren Rechenaufwand als markerbasierte Verfahren.

Grundsätzlich lassen sich auch die featurebasierten Tracking-Verfahren in zwei Gruppen untergliedern, modellbasierte Verfahren, bei denen Versucht wird ein 3D Modell mit dem Kamerabild zu matchen und hieraus die Kamera-Pose zu schätzen und Verfahren, bei denen Features von Bild zu Bild verfolgt werden und aus der Bewegung der Features im Bild auf die Kamera-Pose geschlossen wird.

Während die modellbasierten Verfahren die Kamera-Pose im Koordinatensystem des zu trackenden Objekts berechnen, bewegt sich die Kamera in den Bild zu Bild Verfahren in einem vollkommen unabhängigen Koordinatensystem. In diesem Fall ist die Transformation zwischen den Tracker-Koordinaten und dem Referenz-Koordinatensystem unbekannt, so dass eine Initialisierung mit einem anderen Verfahren notwendig ist, auf dessen Ergebnis das Bild zu Bild Verfahren dann aufsetzen kann.

2.2.1.4 Relative Tracking-Verfahren

Während die obigen Verfahren absolute Positionen und Orientierungen in Relation zu einem festen Koordinatensystem liefern, existieren auch Systeme, die lediglich Werte relativ zu einem zufälligen Startwert bereitstellen können. Hierbei werden mittels Gyroskopen, Accelerometern oder Inklinometern Bewegungen oder Beschleunigungen und die hierdurch verursachten Kräfte gemessen und daraus die relativen Bewegungen des Systems berechnet. Obwohl diese Verfahren für sich alleine genommen nicht für Tracking-Aufgaben in AR Anwendungen geeignet sind, können sie doch in Kombination mit anderen Verfahren zu deutlichen Verbesserungen der Tracking-Ergebnisse führen.

2.2.1.5 Hybride Tracking-Verfahren

Aus den kurzen Beschreibungen der verschiedenen Tracking-Technologien in den vorherigen Abschnitten lässt sich relativ leicht ableiten, dass ein präzises, schnelles Tracking für mobile AR Anwendungen die Verwendung visionbasierter Verfahren erfordert. Aber auch diese Verfahren haben ihre Schwächen, zum Beispiel, wenn nicht genug Features im Bild erkannt werden können oder eine Initialisierung im Referenz-Koordinatensystem erforderlich ist.

Die gezielte Kombination verschiedener Verfahren kann nun dazu genutzt werden, die Schwächen der einzelnen Verfahren zu kompensieren und somit ein besseres Gesamt-Ergebnis zu erzielen. GPS und elektronischer Kompass können beispielsweise in Outdoor Szenarien verwendet werden, um ein visionbasiertes Tracking-Verfahren zu initialisieren. Andersherum kann ein Inertial-Sensor zumindest Orientierungs-Informationen liefern, wenn ein visionbasiertes Verfahren aufgrund von Verdeckungen oder verwischten Kamerabildern kein gültiges Ergebnis bereitstellen kann.

Die Kombination verschiedener Verfahren zu einem hybriden Tracking-System erlaubt es so, für bestimmte Anwendungen oder auch ganze Anwendungsfelder eine Lösung des Tracking-Problems be-

reitstellen zu können. Eine globale Lösung der Tracking-Problematik ist somit allerdings noch nicht erreicht. Nicht zu vernachlässigen in diesem Zusammenhang ist vor allem der Aufwand der Vorbereitung, bis ein solches Tracking-Verfahren in einer bestimmten Anwendung und Umgebung eingesetzt werden kann.

2.3 Präsentation

Augmented Reality bedeutet das Anreichern der realen Umgebung mit zusätzlichen, virtuellen Informationen. Mit dieser Definition sind keine Festlegungen verbunden, in welcher Form die Informationen dem Benutzer präsentiert werden, also welche Wahrnehmungskanäle dabei angesprochen werden. Demgegenüber wurden aber nur wenige Systeme realisiert, bei denen die optische Präsentation nicht im Vordergrund steht. Die Ausnahmen stellen Systeme dar, die abhängig von der Benutzerposition Audio-Annotationen wiedergeben [BEDER95][ROZIER]. Im Folgenden wird entsprechend auch hauptsächlich auf die visuelle Darstellung der virtuellen Informationen eingegangen.

Die visuelle Präsentation von Informationen in AR lässt sich in zwei Aspekte aufteilen, die Hardware, also die verwendeten Displaysysteme und Softwaretechniken für die weitergehende Synthese zwischen realer und virtueller Welt.

2.3.1 Displays

Eine wichtige Rolle bei der visuellen Präsentation der zusätzlichen Informationen in Augmented Reality spielt die verwendete Displaytechnologie. Displays für Augmented Reality Anwendungen können danach kategorisiert werden, wie der Benutzer die reale Welt durch das Display wahrnimmt. Grundsätzlich wird dabei zwischen zwei Modi unterschieden, den optical see through und den video see through Displays.

- *Optical see through Displays* lassen den freien Blick auf die reale Umgebung zu. Nur die virtuellen Szenenanteile werden in das Blickfeld eingeblendet. Die Synthese von realer und virtueller Szene entsteht im semitransparenten Display.
- *Video see through Displays* besitzen keinerlei Transparenz, weshalb der Blick auf die reale Umgebung durch ein Videobild ersetzt werden muss. Zusätzlich zum eigentlichen Display ist somit eine Videokamera notwendig, die das reale Blickfeld des Benutzers aufnimmt. Diese Kamera kann aber zusätzlich auch als Bildquelle für ein visionbasiertes Trackingverfahren verwendet werden.

Vorteil der optical see through Lösung ist, dass der Benutzer die Umgebung direkt und nicht als niedrig aufgelöstes, diskretisiertes Kamerabild wahrnehmen kann. Allerdings stellt eine solche Konstellation auch höhere Anforderungen an das Tracking, da jede Latenz des Tracking-Systems als ein Nachziehen der virtuellen Szene hinter der realen Umgebung wahrgenommen werden kann. Optical see through Displays bieten hier die Möglichkeit, das Videobild so lange zu verzögern, bis das dazugehörige Tracking-Ergebnis vorliegt. Somit wird die Darstellung der kompletten Szene verzögert, was aber als weniger störend empfunden wird, als die dynamischen Registrierungsfehler in optical see through Displays.

Neben der Durchsichtfähigkeit eines Displays gibt es weitere Kriterien, nach denen diese kategorisiert werden können. Ein entscheidendes Merkmal ist dabei ob und wie ein Benutzer ein Display mit sich führt. Die daraus resultierenden Kategorien werden im Folgenden kurz vorgestellt.

- *Head Mounted Displays* (HMD) werden vom Benutzer direkt vor den Augen getragen und bieten dadurch die Möglichkeit, die virtuellen Informationen direkt in dessen Sichtfeld einzublenken. Es sind sowohl optical als auch als video see through HMDs verfügbar. Zusätzlich kann auch zwischen monokularen und biokularen Lösungen gewählt werden. Dabei können biokulare Displays weiter in Mono- und Stereo-Displays unterteilt werden.

Während bei klassischen video see through Displays, ein möglichst großer Blickwinkel das Ziel ist, hat sich eine weitere Displayklasse, die look around Displays herausgebildet, die nur einen sehr kleinen Bereich des Blickfelds abdecken. Der Benutzer kann bei Bedarf darauf fokussieren, andernfalls aber auch einfach daran vorbei schauen. Für AR stellen solche Displays nur eine Notlösung dar, da das Bild der Umgebung nur sehr klein dargestellt werden kann.

- *Handgeführte Displays* können im Mobilen Umfeld eingesetzt werden, wenn es nicht notwendig ist, dass der Benutzer die Hände frei hat. Das Display ist nicht dauerhaft im Blickfeld des Benutzers. Er kann es vielmehr interaktiv vor Objekten der Umgebung platzieren, die für ihn von Interesse sind. Das Display fungiert dabei als eine erweiterte aktive Lupe, der Metapher der magischen Linse [BIER93] folgend. Da es sich fast ausschließlich um video see through Displays handelt, ist hier ebenfalls eine zusätzliche Kamera notwendig, die das Bild der realen Umgebung liefert.

Zur Klasse der handgeführten Displays zählen einfache Flatpanels ebenso, wie TablettPCs und auch PDAs die neben dem Display auch die Recheneinheit beinhalten und zusätzlich über die Möglichkeit der Stifteingabe auch der Interaktion dienen. Eine Mischform zwischen HMDs und handgeführten Displays bilden Binoculars, die bei Bedarf ähnlich einem Fernglas vor die Augen gehalten werden können.

- *Projektor basierte AR Systeme* verfolgen ein gänzlich anderes Konzept. Das Bild wird dabei direkt auf die Umgebung in Blickrichtung des Benutzers projiziert [RASKAR99]. Allerdings funktioniert dies nur dann wirklich gut, wenn auf eine helle, uniforme Oberfläche oder gar retro-reflektierende Materialien projiziert werden kann. Aufgrund der Bauform und des hohen Energieverbrauchs sind aktuell verfügbare Systeme noch nicht für den mobilen Einsatz geeignet. Eingesetzt in stationären Systemen, wie dem Virtual Show Case [BIMB03] lassen sich mit projektorbasierten Systemen allerdings beeindruckende Exponate schaffen.
- *Stationäre Displays* wie Monitore, Flatpanels oder Projektionsdisplays haben für die Realisierung mobiler AR Systeme keine Bedeutung. Sie können aber in verteilten Szenarien, wie dem AR gestützten Remote Experten eingesetzt werden, da auf Seiten des Remote Experten Arbeitsplatzes Mobilität keine Rolle spielt.

Vor allem im Entertainment-Bereich können auch stationäre Projektionsdisplays zum Einsatz kommen. Ein Beispiel hierfür ist der Augmented Man, ein virtueller Charakter der sich zwischen den realen Menschen, die vor der Projektionswand stehen, bewegt und mit diesen interagiert [STRICK00].

2.3.2 Darstellungstechniken

Die grundsätzlichen Darstellungstechniken für AR Anwendungen sind aus der VR Technologie übernommen. So werden auch hier meist szenengraphbasierte Renderer verwendet, die einen hierarchischen Aufbau der darzustellenden Geometrie bieten. Von Seiten der reinen Geometriemenge ist die erforderliche Grafikleistung eines AR Systems gegenüber der eines VR Systems sogar geringer einzuschätzen, da die reale Umgebung nur ergänzt und nicht ersetzt wird. Andererseits stellt gerade die Kombination zwischen realer und virtueller Umgebung eine neue Herausforderung dar. Die dabei entstehenden Probleme können je nach Anwendung in zwei Kategorien eingeteilt werden, entweder den Eindruck des Verschmelzens von real und virtuell zu verstärken oder aber virtuelle Informationen besonders hervorzuheben.

Die Techniken zur Verbesserung der Synthese aus real und virtuell wirken alle auf ein physikalisch plausibleres Verhalten der virtuellen Objekte hin. Die dabei im Rendering zu berücksichtigenden Interaktionen der virtuellen Objekte mit ihrer realen Umgebung sind realistische Beleuchtung sowie die korrekte Darstellung von Verdeckungen zwischen realen und virtuellen Objekten. Die Anpassung der Beleuchtung der virtuellen Objekte an die realen Beleuchtungsverhältnisse erfordert zwei Schritte, die Analyse der realen Beleuchtung und deren Rekonstruktion in der virtuellen Szene. Da dies aber sehr aufwändig und nicht in Echtzeit durchführbar ist, beschränkt man sich hierbei auf den Schattenwurf zwischen realen auf virtuellen Szenenteilen [HALL03][SUGA03]. Auch der vorherige Schritt der Beleuchtungsrekonstruktion kann meist vernachlässigt werden, da Schatten meist als depth-cue, als Hinweis auf die räumliche Einordnung eines Objekts in einem 2D Bild verwendet werden. Den gleichen Zweck erfüllt auch die Berücksichtigung der korrekten Verdeckung zwischen realer Szene und virtuellen Objekten [WLOKA95].

Je nach Anwendung ist mehr oder weniger visueller Realismus gefordert. Die Darstellung der Arbeitsschritte einer Serviceanleitung soll vor allem die Aufmerksamkeit des Benutzers auf einen Punkt der realen Umgebung steuern, es kommen meist kräftige Signalfarben zum Einsatz, die sich stark vom realen Hintergrund abheben. Im Gegensatz dazu erfordert die Beurteilung eines virtuellen Prototyps oder die Präsentation der Rekonstruktion antiker Tempelanlagen ein hohes Maß an visuellem Realismus. Das betrifft die realistische Wiedergabe der Materialien des virtuellen Objekts ebenso, wie dessen Interaktion mit der realen Umgebung. Herauszuheben ist dabei die korrekte Darstellung von Verdeckungen zwischen realen und virtuellen Objekten sowie die Berücksichtigung der realen Lichtsituation.

Die Präsentation von visuellen Informationen in einer virtuellen Umgebung kann abhängig vom Referenzsystem, in dem sie fixiert sind, kategorisiert werden. Billinghurst unterscheidet dabei zwischen drei Modi [BILL98]:

- *Head-stabilized* – Die Positionierung der Informationen erfolgt im Bildraum. Unabhängig davon, wo der Benutzer gerade hinschaut, haben die Informationsobjekte eine fixe, display-relative Position. Der Informationsraum entspricht genau der Bildfläche des Displays. Es ist kein Tracking notwendig.
- *Body-stabilized* – Informations-Objekte besitzen eine fixe Position relativ zum Körper des Benutzers. Der Benutzer kann den sichtbaren Bereich des so gebildeten Raumes durch Kopfbewegungen

onen ändern. Hierfür ist es notwendig, dass die Kopforientierung relativ zum Körper erfasst wird. Der Informationsraum entspricht je nach Realisierung einer Kugel oder einem Zylinder um den Benutzer herum. Position und Orientierung des Informationsraumes sind fix zum Körper des Benutzers. Die Navigation in diesem Raum erfolgt typischerweise über die Änderung der Blickrichtung des Anwenders, kann aber alternativ auch über andere Interaktionstechniken gesteuert werden.

- *World-stabilized* – Die Position und Orientierung der Informationsobjekte ist mit der realen Umgebung fest verbunden, die Objekte sind mit der realen Welt registriert. Voraussetzung ist, dass Position und Blickrichtung des Benutzers erfasst werden, so dass für ihn die Illusion entsteht, dass reale Welt und virtuelle Objekte eine Einheit bilden.

Nach Azumas Charakterisierung von AR ist eine grundlegende Eigenschaft die Registrierung der virtuellen Objekte mit der realen Umgebung. Daraus lässt sich allerdings nicht ableiten, dass die Darstellung der Objekte zwangsläufig world-stabilized erfolgen muss. Vielmehr können Mischformen der Präsentation genutzt werden, die Darstellung übersichtlicher zu gestalten und die Interaktion zu vereinfachen. Ein Text-Label, das aus Gründen der Lesbarkeit display-stabilized dargestellt wird, kann über eine Linie mit einem Punkt der Umgebung verbunden werden, um so den räumlichen Bezug darzustellen. Darüber hinaus können die Objekte im Bildraum so organisiert werden, dass alle wichtigen Informationen, real und virtuell, gleichzeitig sichtbar sind [BELL01]. Sollen allerdings virtuelle 3D Geometrien in die reale Umgebung integriert werden, so ist eine Darstellung im world-stabilized Modus unabdingbar.

2.3.3 Interaktion

In Augmented Reality kann der Benutzer mit seiner realen Umgebung und virtuellen Objekten zur gleichen Zeit interagieren. Hieraus lässt sich die Anforderung ableiten, dass die Interaktion mit den virtuellen Objekten möglichst genau der Interaktion mit realen Objekten entsprechen sollte. Im Kontext von virtuellen Umgebungen spricht man in diesem Zusammenhang davon, dass die Interaktion möglichst natürlich gestaltet sein sollte. Im Idealfall sollte für den Anwender kein Unterschied in der Interaktion mit realen und virtuellen Objekten zu erkennen sein, reale und virtuelle Szene würden auch auf der Ebene der Interaktion eine perfekte Synthese eingehen.

Aus heutiger Sicht, mit den bestehenden technologischen Möglichkeiten, ist dies allerdings Utopie. Die Erfahrungen auf dem Gebiet der Interaktion mit immersiven virtuellen Umgebungen haben gezeigt, dass es bereits eine schwierige Aufgabe darstellt, intuitive Interaktionstechniken in diesen Umgebungen bereit zu stellen. Auch wenn VR in einigen Bereichen als Werkzeug akzeptiert und eingesetzt wird, können diese Systeme meist nur von Experten bedient werden.

Gegenbeispiel sind hier öffentliche VR Installationen im Bereich Entertainment und Edutainment, die eine speziell auf das jeweilige Exponat zugeschnittene Interaktion besitzen. Ein herausragendes Beispiel hierfür ist das virtuelle Graffiti, bei dem der Benutzer eine nachgebildete Spraydose als Interaktionsgerät bedient. Diese Dose verhält sich so, wie man es von einer realen Spraydose verhält, nur dass sie digitale Farbe auf ein Projektionsdisplay aufträgt. Diese Benutzerschnittstelle ist fast ohne Erklä-

rungen sofort verständlich, allerdings ist diese Lösung so spezialisiert, dass sie quasi nicht auf andere Anwendungen übertragen werden kann.

2.3.3.1 Interaktionsgeräte in immersiven VR Umgebungen

Abgesehen von einigen exotischen, anwendungsspezifischen Lösungen wurden diverse Interaktionsgeräte entwickelt, die allgemeiner einsetzbar sind aber trotzdem eine intuitive Bedienung von VR Anwendungen erlauben sollen. Ein in sechs Freiheitsgraden (6DOF) getracktes Eingabegerät erlaubt die präzise Steuerung eines Selektionsstrahls oder die intuitive Transformation eines selektierten Objekts im Raum. In VR Umgebungen wird das Tracking der Interaktions-Geräte üblicherweise über fest installierte elektromagnetische, ultraschallbasierte oder visionbasierte Tracking-Systeme realisiert, die einen recht kleinen Arbeitsbereich abdecken. Die so realisierten, nicht tischgebundenen Eingabegeräte werden auch als fliegende Geräte zusammengefasst, die erst ein Arbeiten in immersiven Umgebungen ermöglichen. Neben der Sensorik für das verwendete Tracking-System enthält das Interaktionsgerät zusätzlich ein oder mehrere Buttons die der Triggerung von Interaktionsaufgaben dienen. Beispiele für solche Geräte sind Flightstick [KNÖ03], Bat [WARE88] oder Wand [BRED99]. Auch das Personal Interaction Panel (PIP) des Studierstube-Systems wurde in seiner ursprünglichen Form basierend auf einem 6 DOF Tracking-System entwickelt [SZAL97]. Dieses an sich passive Tablett wird durch die Einblendung virtueller GUI-Elemente und deren Bedienung über den getrackten Pen zu einem recht universellen Eingabegerät. Zur unabhängigen Erfassung von Tablett und Stift sind hier aber sogar schon zwei unabhängige Tracking-Sensoren notwendig.

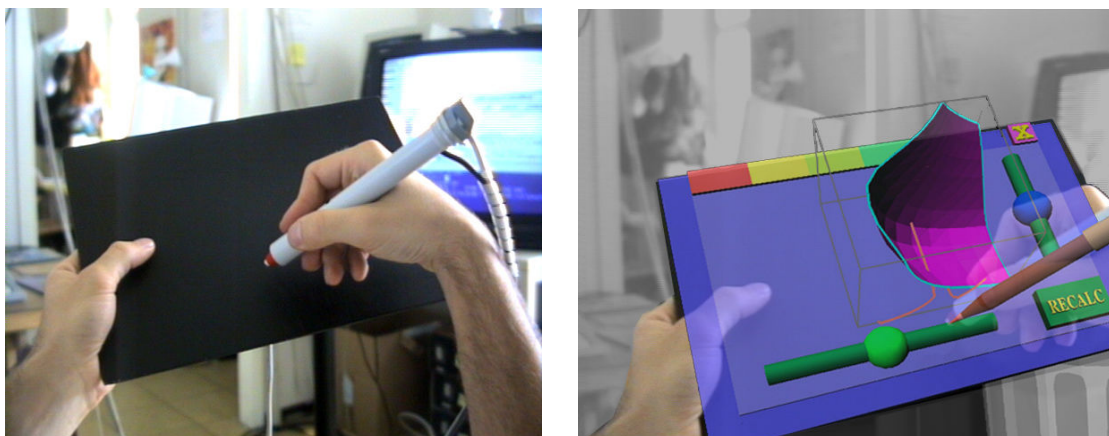


Abbildung 2: Personal Interaction Pad (PIP)

Vielfach wird Interaktion in VR direkt in Verbindung mit Datenhandschuhen für die Eingabe über Gesten gebracht [BÖHM96]. Der Datenhandschuh erfasst die Hand- und Fingerstellung des Anwenders, woraus wiederum die Gesten interpretiert werden können. Das Erkennen einer Geste löst dann eine Interaktion aus. Da Datenhandschuhe meist in Verbindung mit einem 6DOF Tracking-System verwendet werden, kann der Benutzer über die Position und Orientierung seiner Hand zusätzlich Selektions- und Manipulationsaufgaben durchführen.



Abbildung 3: Datenhandschuh

Ebenfalls ein typisches Eingabegerät für VR Systeme ist die Spacemouse [MAGE], bei der es sich um ein tischgebundenes Eingabegerät handelt, das beidhändig auch losgelöst von einer Tischoberfläche verwendet werden kann. Die Spacemouse erlaubt die Eingabe in 6 Freiheitsgraden und bietet zusätzlich eine Reihe freiblegbarer Tasten. Für den dauerhaften, nicht tischgebundenen Einsatz ist die Spacemouse aber zu schwer und benötigt zum präzisen Positionieren viel Aufmerksamkeit und Übung.



Abbildung 4: Spacemouse

2.3.3.2 Interaktionsgeräte für mobile Anwendungen

Bedingt durch die typischen Anwendungen, wie Datenerfassung, die auf mobilen Rechnern durchgeführt werden, orientieren sich auch die Eingabegeräte für solche Geräte an den Desktopfunktionalitäten, mit denen eine Anwender meist bereits vertraut ist. Oft werden auf solchen Rechnern auch vereinfachte Varianten der bekannten Desktopbetriebssysteme betrieben, die wie das Desktop-Pendant über eine WIMP (Window, Icon, Mouse, Pointer) Benutzerschnittstelle bedient werden.

Wie auch am Desktop handelt es sich bei den Interaktionsgeräten überwiegend um Tastaturen und 2D-Pointinggeräte, die Mouse-Funktionalitäten zur Verfügung stellen aber speziell an die Gegebenheiten des mobilen Einsatzes angepasst sind.

Für die Eingabe alphanumerischer Werte in mobile Rechner stehen sehr unterschiedliche Eingabegeräte zur Verfügung. Hier kann unterschieden werden zwischen Designs, die an klassische Tastaturen angelehnt sind, sogenannten Chordic Keyboards, bei denen Zeichen aus der Kombination mehrerer Taste, den Chords, gebildet werden und Eingabemöglichkeiten für virtuelle Tastaturen. Die *WristPC* Tastatur [WRISTPC], wird am Unterarm getragen und mit der Hand des anderen Arms bedient. Ob-

wohl die Anzahl der Tasten stark reduziert ist, besitzt die WristPC Tastatur ein relativ konservatives Tasten-Layout. Der *Twiddler2* [TWID2] ist eine Kombination aus einem Chordic Keyboard und einem Trackpoint Mouse-Pointer. Die Tastatur besitzt 12 Finger Tasten und 6 weitere zur Bedienung mit dem Daumen. Jede Kombination dieser Tasten kann ein Kommando oder ein Zeichen erzeugen. Wird der Mouse-Pointer mit dem Daumen bewegt, fungieren die Tasten auf der Frontseite als Maustasten. Das *FrogPad* [FROG] ist eine Tastatur, die für die einhändige Bedienung entworfen wurde. Das spezielle Tastenlayout soll dabei ein besonders schnelles Erlernen der Eingabe von Text ermöglichen.



Abbildung 5: WristPC Unterarmtastatur, Twiddler2 Chordic-Keyboards und FrogPad

Auf WIMP Benutzerschnittstellen wie Apples Mac OS™ oder Microsoft Windows™ spielt die Maus als Eingabegerät eine herausragende Rolle. Applikationen werden über die Maus gestartet, Dateien verschoben und Menüpunkte ausgewählt. Jeder, der mit der Bedienung einer solchen Benutzerschnittstelle vertraut ist, ist auch mit der Mausmetapher vertraut.

Bei der klassischen Maus ist eine recht geringe Reiz-Reaktions-Korrespondenz zu beobachten. Eine Bewegung in der horizontalen Tischebene nach vorne wird in eine vertikale Bewegung des Cursors nach oben umgesetzt. Das Prinzip der Maus ist, dass sie auf einer ebenen Unterlage transliert und diese Bewegung meist über die optische Abtastung der Unterlage erfasst wird. In mobilen Systemen steht diese Unterlage im Allgemeinen nicht zur Verfügung. Die Cursor-Bewegungen müssen hier also anders gesteuert werden. Im Folgenden werden nun einige Geräte vorgestellt, die eine 2D Mausfunktionalität auch in mobilen Systemen zur Verfügung stellen.

Bei der *GyroRemote* werden die translatorischen Bewegungen des Mauscursors über Drehungen der Hand erzeugt. Eine Drehung der Hand nach links bewirkt dabei eine Bewegung des Cursors nach links. Analog erfolgt die Abbildung von Handdrehungen in den anderen Richtungen. Im Gegensatz zur Desktop Maus werden vertikale Mausbewegungen auch durch vertikale Handbewegungen erzeugt. Der eigentliche Verwendungszweck des Gerätes ist die Steuerung von Präsentationen mit der Möglichkeit auch den Mauszeiger auf intuitive Art steuern zu können. Da das Eingabegerät kabellos über Funk mit dem Rechner kommuniziert, ist es sehr gut für den mobilen Einsatz geeignet. Allerdings verlangt vor allem das präzise Ansteuern kleiner Bedienelemente auf einer graphischen Oberfläche viel Übung und eine ruhige Hand. Vor allem das Verharren in der Zielposition und gleichzeitige Drücken einer der Tasten erweist sich dabei als relativ schwierig. Beim *Handtrack* handelt es sich um einen kleinen Trackball, bei dem die Kugel mit dem Daumen bewegt wird. Der Handtrack ist speziell für den Einsatz mit mobilen Rechnern konzipiert und kann ebenso wie die GyroRemote mit einer Hand bedient werden. Besonders Cursorbewegungen über längere Distanzen sind allerdings relativ umständlich zu erreichen, da hierbei mehrfach angesetzt werden muss.



Abbildung 6: GyroRemote, Handtrack

Das Interaktionsgerät Mike (siehe Abbildung 7) wurde am Fraunhofer Institut für Arbeitswirtschaft und Organisation (IAO) für die Interaktion vor immersiven Großdisplays entwickelt. Dieses Gerät erlaubt die drahtlose Messung der Hand-Neigung mittels Accelerometern. Im Gegensatz zu Gyroskop basierten Geräten kann so die absolute Neigung gemessen werden.

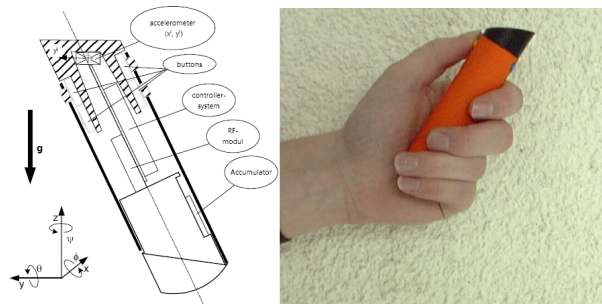


Abbildung 7: Interaktions-Gerät Mike

Da Mike speziell für die Navigation in virtuellen Umgebungen ausgelegt wurde, ist die Anbindung im VR System auch dementsprechend gestaltet. Die Abweichung in der Neigung von der Nulllage um die zwei verfügbaren Achsen des Gerätes kann dort auf unterschiedliche Attribute wie zum Beispiel Geschwindigkeit abgebildet werden.

2.4 Augmented Reality basierte Assistenzfunktionen

Der Begriff Augmented Reality wurde von Caudell und Mizell [CAU92] bei der prototypischen Realisierung eines Systems zur Unterstützung bei der Flugzeugmontage in den frühen 90er Jahren geprägt. Ein Monteur steht bei dieser Aufgabe typischerweise vor einer mehrere Meter langen Tafel, auf der die zu erstellenden Kabelverbindungen aufgedruckt sind. Von jeder Variante der Kabelbäume muss eine solche Tafel vorhanden sein. Caudell und Mizell verwenden stattdessen eine neutrale Tafel und blenden die Anweisungen, wo die nächste Verbindung herzustellen ist, in einem durchsichtigen Head Mounted Display (HMD) ein (Abbildung 8). Damit die Montageanweisungen auch tatsächlich an der richtigen Stelle im Blickfeld des Benutzers erscheinen ist das HMD getrackt, d.h. Position und Orientierung werden erfasst, so dass der Computer, der die virtuelle Darstellung bereitstellt, immer weiß, wo der Benutzer gerade hinsieht. In der prototypischen Anwendung, bei der nur ein relativ begrenzter

Arbeitsraum vom Tracking abgedeckt werden muss, wird ein elektromagnetisches Tracking-System eingesetzt.

Ziel war die schnelle Realisierung eines Demonstrators, weshalb hier auf Hardwareseite nur marktverfügbare Geräte Verwendung fanden. Trotz verschiedener technischer Unzulänglichkeiten, vor allem der Hardware, gelang es mit diesem System das Potential der Technologie zu demonstrieren. Es zeigte sich allerdings auch, dass die Zeit für operativ einsetzbare AR Systeme noch nicht reif war.



Abbildung 8: Augmented Reality unterstütztes Zusammenstellen von Kabelbäumen [Boeing]

Dass der Einsatz von Augmented Reality Unterstützung von Montageaufgaben belegbare Vorteile bringt, zeigen unterschiedliche Experimente. Tang et al [TANG03] untersuchten die Effektivität von AR Unterstützung im Vergleich zu gedruckten Anleitungen und computergenerierten Informationen (CAI) präsentiert auf einem LCD-Panel und in einem See-Through-HMD. Alle computergestützten Methoden zeigen einen Zeitgewinn gegenüber der gedruckten Dokumentation, allerdings kann kein signifikanter Gewinn von AR gegenüber den anderen computergestützten Methoden nachgewiesen werden (Abbildung 9 links).

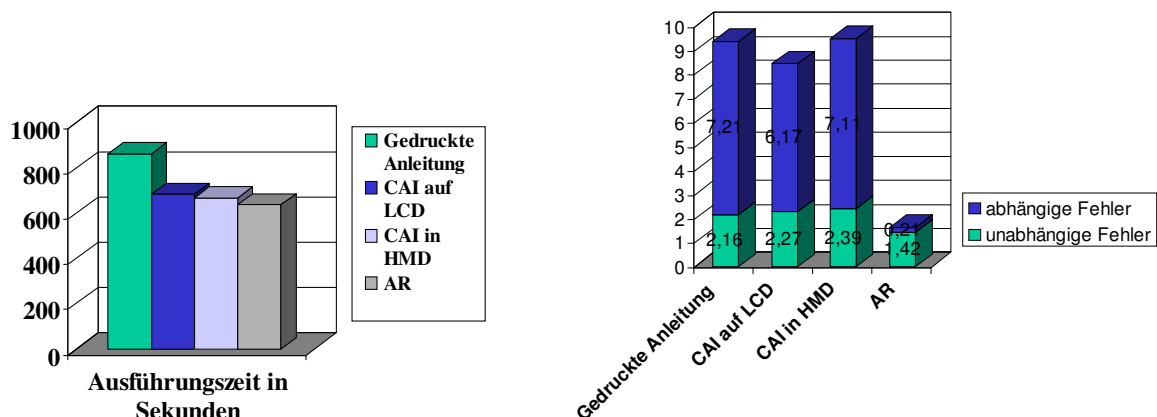


Abbildung 9: Performanzvergleich verschiedener Anleitungsformen

Bei der Betrachtung der Fehlerquote allerdings bringt der Einsatz von Augmented Reality einen deutlichen Gewinn gegenüber allen anderen untersuchten Methoden (Abbildung 9 rechts).

Typischerweise lässt sich die Durchführung einer Servicetätigkeit in Informationsbeschaffung und die eigentliche manuelle Tätigkeit gliedern, wobei der zeitliche Anteil der Beschaffung der notwendigen Informationen bei Reparaturen an Flugzeugen bis zu 45 % beträgt [NEUM98]. Die Bereitstellung kontextabhängiger Daten und deren Präsentation in räumlicher Relation zu den realen Objekten im Blickfeld des Benutzers können die Dauer der Suche nach den notwendigen Informationen drastisch verkürzen. Zusätzlich fördert die räumliche Darstellung der Informationen das Erstellen eines mentalen Modells beim Anwender.

Ein frühes Beispiel einer Augmented Reality basierten Assistenzfunktion bei der Durchführung einer Servicetätigkeit beschreibt die Wartung eines Laserdruckers [FEIN93]. Die einzelnen Wartungsschritte werden als animierte Drahtgittermodelle in einem durchsichtigen HMD dargestellt (Abbildung 10). Das AR System verfolgt sowohl den Benutzer als auch Objekte der Umgebung über ein verteiltes Ultraschall-Tracking-System. Jedes Objekt, das von diesem System erfasst werden soll, muss zu diesem Zweck mit einem Ultraschalltransmitter versehen werden.



Abbildung 10: AR unterstützte Wartung eines Laserdruckers [UNC]

Ein weiteres Anwendungsgebiet, für das bereits früh das Potential von Augmented Reality gestützten Assistenzfunktionen erkannt wurde, ist die Medizin. Patientenspezifische Datensätze die sowohl prä- als auch intra-operativ gewonnen werden, können in das Blickfeld eines Operateurs eingeblendet werden. Vor allem in der minimal invasiven Chirurgie erhält der Arzt so Einblicke und Orientierungshilfen während einer Operation, die ihm ansonsten nicht in dieser Form zur Verfügung stehen. Das Ergebnis ist ein, um weitere sinnvolle Informationen angereicherter, Röntgenblick in das ansonsten verborgene Innere des Patienten. Ein Beispiel für eine solche Anwendung ist die lagerichtige Anzeige von aktuellen Ultraschalldaten [BAJU92][SATO98].

2.5 Benutzerschnittstellen

Der Begriff Benutzerschnittstelle beschreibt alle Komponenten und Aspekte eines Mensch-Computer-Systems, mit denen ein Benutzer begrifflich oder sensumotorisch in Verbindung kommt [MORA81].

Dazu gehören auch werkzeugunabhängiges Aufgabenwissen und werkzeugspezifisches Wissen des Benutzers, also auch das Wissen über Computersysteme und kognitive und sensumotorische Fertigkeiten zu dessen Benutzung [WAND93].

Eine zentrale Rolle bei der Gestaltung von Benutzerschnittstellen für Computer und Software spielen die kognitiven Prozesse, die beim Benutzer selbst ablaufen. Eine effiziente Benutzerschnittstelle hat sich hier zuallererst an den Bedürfnissen und Voraussetzungen des Benutzers zu orientieren. Ein Großteil der hier bisher gewonnenen Erkenntnisse bezieht sich allerdings auf Desktopsysteme und die dort üblicherweise vorzufindenden Aufgaben und Benutzerschnittstellen. Zur Gestaltung von Benutzerschnittstellen für VR und besonders AR gibt es dagegen nur wenige, punktuelle Untersuchungen.

Die Aufgabe besteht nun darin, zu analysieren, welche Erkenntnisse der Software Ergonomie, die für Desktopsysteme oder auch VR gewonnen wurden auch auf Augmented Reality übertragen oder adaptiert werden können. Hier sind vor allem die Prozesse zu nennen, die im Benutzer selbst ablaufen und somit unabhängig von der Anwendung und der System-Konfiguration sind.

Der Computer und die verwendete Software sind als Werkzeug zu verstehen und sollten somit wie reale Werkzeuge optimal an die jeweilige Aufgabe angepasst sein. Obwohl ein Hammer sicherlich für viele Aufgaben geeignet ist, gibt es sicherlich mehr Situationen, in denen es besser geeignete Werkzeuge gibt. Auch ein falsch dimensionierter Hammer kann ungeeignet sein, wenn es gilt Nägel einer bestimmten Größe einzuschlagen. Man spricht in diesem Zusammenhang auch von der Aufgabenangemessenheit. Vor der Wahl des richtigen Werkzeugs muss demzufolge eine genaue Analyse der Aufgabe stehen. Dies ist umso schwieriger, wenn es sich bei dem zu entwickelnden System um ein generisches Rahmensystem handelt. Helfen kann hier die Modellierung der Prozesse und Eigenschaften, die alle Benutzer und Anwendungen gemeinsam haben.

Ein wichtiger Aspekt bei der Betrachtung von Benutzerschnittstellen ist die menschliche Wahrnehmung. Diese hat starken Einfluss auf die Prozesse der Informationsverarbeitung beim Benutzer. Da der Benutzer die Umgebung mit seinen Sinnesorganen wahrnimmt, sind zu erst die durch sie vorgegebenen Rahmenbedingungen festzuhalten. Beim Umgang mit einem Computersystem sind dies vor allem der visuelle, der auditive und der taktile Sinn. Aufgrund der dominanten Bedeutung des visuellen Sinnes bei der Aufnahme von Informationen wird im Folgenden hauptsächlich auf die Beschaffenheit des visuellen Systems eingegangen.

Der Mensch nimmt Licht in einem für ihn sichtbaren Wellenlängenbereich über die *Retina* seiner Augen auf. Dort können Intensität, Wellenlänge und räumliche Verteilung von entsprechenden Rezeptoren aufgenommen werden. Dabei wird die Retina in einen kreisförmigen Bereich scharfen Sehens, die *Fovea*, und den umgebenden Bereich *peripheren Sehens* unterteilt. Die Fovea deckt dabei nur einen Bereich mit einem Durchmesser von 4° ab. Durch Augenbewegung wird ein Objekt in den fovealen Bereich gebracht. Dieser Vorgang nennt sich die *Fixation* eines Objekts.

Die Augenbewegungen zur Fixierung eines Objekts nennt man auch *Sakkaden*. Die Dauer einer solchen Sakkade allein beträgt dabei 30 ms, die Zeit für die komplette Fixation im Mittel 230 ms [CARD83]. Die Sakkaden können Objekte erreichen, die bis zu 30° von der Richtung des vorherigen Fixationspunkts differieren. Liegt ein Objekt außerhalb dieses Bereichs, sind zusätzlich Kopfbewegungen notwendig, um es zu erfassen.

Die aufgenommenen visuellen Reize werden im so genannten ikonischen Speicher abgelegt. Reizwerte werden hier noch nicht symbolisch oder begrifflich, sondern vielmehr in Form analoger Rohdaten gehalten. Die Speicherdauer beträgt ca. 200 ms bevor ein langsames Verblässen einsetzt, das aber durch erneutes Auffrischen des Reizes verhindert werden kann.

Das visuelle System ist in der Lage Reize bis zu einer Grenze von 100 ms zeitlich aufzulösen [CARD83]. Werden innerhalb der zeitlichen Auflösung mehrere ähnliche Reize auf den gleichen Ort der Retina projiziert, werden diese zu einem einzigen Wert integriert. Zwei ähnliche Reize an unterschiedlichen Orten in einem Abstand von 100 ms führen zur Wahrnehmung einer Bewegung des Reizes zwischen den beiden Orten, der so genannten Scheinbewegung.

Die Zeichen- und Figurerkennung geschieht durch Interpretation der im ikonischen Speicher vorliegenden Reize. Ergebnis ist eine symbolische oder begriffliche Repräsentierung im Kurzzeitgedächtnis (KZG). Erst jetzt, da der Reiz in einer Form zur willkürlichen Weiterverarbeitung vorliegt, kann man auch von einer Wahrnehmung des Gegenstands sprechen.

Die Steuerung der Auswahl von Inhalten des ikonischen Speichers für die Erkennung erfolgt in der Regel automatisch, kann aber auch über kognitive Prozesse des Kurzzeitgedächtnisses gesteuert geschehen. Die Auswahl basiert dabei auf Merkmalen wie Intensität, Farbe, Kontrastverteilung und anderen physikalischen Merkmalen.

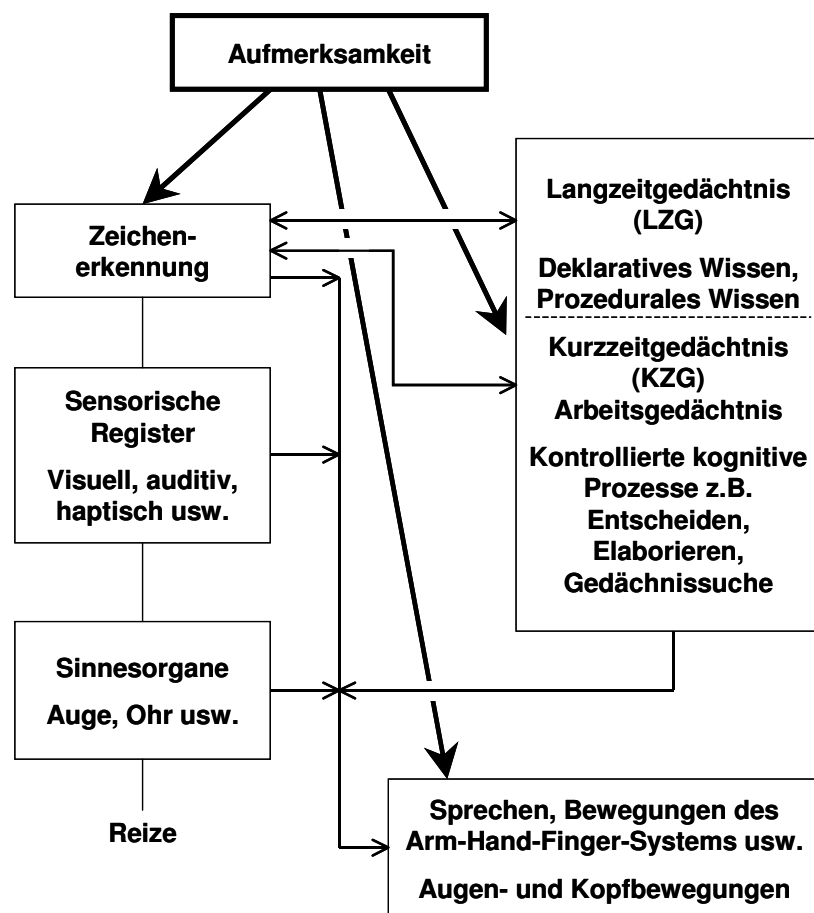


Abbildung 11: Funktionale Einheiten der menschlichen Informationsverarbeitung und Handlungssteuerung [WAND93]

Eine zentrale Rolle bei der menschlichen Informationsverarbeitung spielt das Kurzzeitgedächtnis (KZG). Seine Funktion ist das kurzzeitige Speichern einer begrenzten Informationsmenge als Arbeitsspeicher und Ort kognitiver Prozesse [WAND93].

Die Informationen im KZG sind symbolische Repräsentationen von Wahrnehmungen oder Erinnerungen aus dem Langzeitgedächtnis. Die Inhalte des KZG können als Argumente für Kognitive Prozesse verwendet werden. Sie sind unmittelbar verfügbar und bewusst. Die Kapazität des KZG ist dabei aber sehr begrenzt und liegt bei etwa 3 bis 4 Einheiten [CARD83]. Die Komplexität der einzelnen Einheiten kann dabei stark variieren, zwischen einzelnen Buchstaben und Ziffern, über Teile von Sätzen bis zu komplexen visuellen Vorstellungseinheiten. Limitierend ist also die Anzahl der Speicherplätze, weniger deren Informationsgehalt.

Ziel bei der Präsentation von Informationen sollte also sein, möglichst wenige Objekte mit jeweils hohem Informationsgehalt darzubieten, um so den Inhalt des KZG effizient zu steuern. Allerdings lässt sich in Augmented Reality Anwendungen nur der Informationsgehalt der virtuellen Szenenanteile steuern. Die Informationsfülle der realen Umgebung kann dagegen nicht kontrolliert werden.

Reiz-Reaktions-Korrespondenz (engl. *Stimulus response correspondence* oder auch *stimulus response compatibility*) [FOL90]. Eine Korrespondenz von Reiz und Reaktion liegt vor, wenn auf einen Reiz, z.B. eine Handbewegung nach links, eine Objektbewegung in dieselbe Richtung erfolgt. Bei allen Interaktionen mit realen Gegenständen in der realen Welt korrespondiert Reiz und Reaktion maximal-optimal. Beim Arbeiten mit 2D Benutzungsoberflächen und der 2D-Maus hingegen liegt keine Reiz-Reaktions-Korrespondenz vor, denn der Mauszeiger bewegt sich auf dem Bildschirm nach oben, wenn die Maus vom Benutzer weg 'nach hinten' bewegt wird. Liegt keine Reiz-Reaktions-Korrespondenz vor, so ist eine mentale Abbildung (*mental mapping*) seitens des Benutzers notwendig, um zur gewünschten Aktion die korrespondierende Bewegung auszuführen. Für dreidimensionale Aufgaben, wie etwa die Manipulation eines 3D-Modells, kann eine maximale Reiz-Reaktions-Korrespondenz nur mit 3D Eingabegeräten erzielt werden. Ansonsten entsteht bei der Bearbeitung durch den Anwender erhöhter Aufwand bei der Bewältigung mentaler Abbildungsvorgänge. Dieser Aspekt besitzt vor allem bei der Unterstützung realer Aufgaben mittels AR große Bedeutung, da hier jede zusätzliche mentale Belastung die Konzentration auf die eigentliche, reale Aufgabe verringert.

Bei der Gestaltung eines Interaktionsgerätes ist die Ergonomie als Anforderung mit hoher Priorität zu betrachten. Die Belastung des Benutzers durch das zusätzliche Interaktionsgerät sollte minimal sein, dies betrifft sowohl die körperliche als auch die geistige Belastung.

Körperliche Belastung beim Führen eines Interaktionsgerätes äußert sich darin, dass der Benutzer Muskelkraft aufwenden muss, um mit dem System zu kommunizieren. Hier kann unterschieden werden in der Belastung großer Muskelgruppen, wie beim Heben des Armes, und kleiner Muskelgruppen zum Beispiel durch Bewegungen der Hand oder einzelner Finger. Werden die großen Muskelgruppen beansprucht, führt dies zu einer schnelleren Ermüdung des Anwenders. Aber auch, wenn nur die kleineren Muskelpartien des Unterarms oder der Hand beansprucht werden, ist unnötige Belastung durch das hohe Gewicht eines Eingabegerätes zu vermeiden. Dieser Aspekt ist vor allem dann von Bedeutung, wenn neben der Haltearbeit auch das präzise Ausführen von feinen Bewegungen während der Eingabe gefordert ist.

Interaktionstechniken, bei denen der Benutzer eine natürliche, für ihn bequeme Haltung einnehmen kann, sind solchen, bei denen er eine unbequeme Haltung einnehmen muss, vorzuziehen. Ein Beispiel hierfür sind Selektionstechniken, bei denen beide Hände in das Blickfeld des Anwenders gehalten werden müssen.

Wie vorher kurz angesprochen, besitzt die ergonomische Gestaltung eines AR Assistenz Systems hohe Priorität, da die helfende Funktion hier nur dann vom Anwender Angenommen wird, wenn er sie nicht als zusätzliche Belastung empfindet. Dies fängt beim Gewicht der verwendeten Hardware an, vor allem dem Gewicht und dessen Verteilung, dass er über ein HMD am Kopf tragen muss. Das geht weiter über die Qualität der Darstellung, geprägt durch die Qualität des Displays und die Art der gewählten Darstellung bis zur Gestaltung der Interaktion mit dem System, die vor allem durch die gewählten Eingabegeräte und –Techniken geprägt wird.

2.6 Zusammenfassung

Augmented Reality (AR) erlaubt es, die reale Umgebung eines Anwenders mit zusätzlichen, virtuellen Informationen anzureichern. Dabei bleibt die korrekte räumliche Relation zwischen realer und virtueller Umgebung unabhängig von den Bewegungen des Anwenders erhalten. Es entsteht so der Eindruck, dass reale und virtuelle Welt eine Einheit bilden. Voraussetzung für diesen Eindruck ist, dass Position und Blickrichtung des Anwenders präzise und schnell erfasst werden. Hierfür bieten sich besonders in mobilen AR Anwendungen visionbasierte, markerlose Tracking-Verfahren an, da diese keine Instrumentierung der Umgebung benötigen. Um die Nachteile dieser Verfahren zu kompensieren, werden sie in Verbindung mit anderen Verfahren in hybriden Tracking-Systemen eingesetzt.

Neben dem Tracking stellen Präsentation und Interaktion zwei weitere Grundfunktionalitäten eines AR Systems dar. Neben unterschiedlichen Displaytechnologien sind bei der Präsentation von AR Inhalten auch unterschiedliche Darstellungsmodi zu betrachten, die je nach Anwendung und Hardwarekonfiguration ihre spezifischen Vor- und Nachteile mitbringen.

Die Interaktion wird vor allem durch die einsetzbaren Geräte geprägt. Eine Betrachtung von marktverfügbaren Eingabegeräten ergibt, dass diese ein erhebliches Defizit entweder an Funktionalität für 3D Interaktion aufweisen oder aber 3D Interaktion durch spezielle Tracking-Hardware, die im mobilen Einsatzumfeld nicht verfügbar ist, realisieren. Für die Kombination von 3D Interaktion in der an sich dreidimensionalen AR Umgebung gilt es also eigene Hardware zu entwickeln und dabei neue Ansätze zu verfolgen.

Wie lohnend die Bemühungen sind, AR Technologien einfach und vor allem auch im mobilen Umfeld nutzbar zu machen, zeigen einige Untersuchungen über das Potential von AR basierten Anwendungen. Speziell AR basierte Assistenzfunktionen, versprechen durch den Einsatz von AR Techniken erhebliche Performanzsteigerungen.

Zur zielgerichteten Erstellung und Gestaltung von AR Anwendungen ist neben der Kenntnis der technischen Randbedingungen vor allem auch Kenntnis der physiologischen Voraussetzungen der potentiellen Anwender notwendig. Die Gestaltung von Präsentations- und Interaktionstechniken sollte vor allem den Mensch im Mittelpunkt der Betrachtung haben. Nur wenn dessen Bedürfnisse optimal un-

terstützt und seine Möglichkeiten ebenso optimal genutzt werden, kann die notwendige Akzeptanz des Anwenders erreicht werden.

3 Anforderungen an ein mobiles AR System

Neben der Betrachtung der Anforderungen an ein Laufzeitsystem, soll der Gesamtprozess, also auch die Schritte der Vorbereitung des Einsatzes von AR Technologien im jeweiligen Bereich betrachtet und den Prozessen, die bei der Vorbereitung herkömmlicher Unterstützungssysteme auftreten, gegenübergestellt werden. Bei der Vorbereitung des Einsatzes von AR Technologien in unterschiedlichen Anwendungsgebieten ist vor allem zu betrachten, welche Daten und Informationen benötigt werden, hier vor allem für das Tracking oder zur Präsentation. Zu unterscheiden bleibt dann, je nach spezifischer Anwendung, welche der Daten bereits vorliegen und eventuell in ein geeignetes Format transformiert werden müssen oder welche Daten erst noch erstellt werden müssen.

Weitere Faktoren, die kritisch betrachtet werden müssen, ist der Aufwand, der zur Bedienung eines Systems zur AR basierten Unterstützung benötigt wird und wie groß der Aufwand ist, sich mit der Bedienung eines solchen Systems vertraut zu machen.

3.1 Anwendungsgebiete

Alleine schon aufgrund der hohen Hardwareanforderungen beschränkt sich der Einsatz von immersiven VR Installationen auf relative kleine Bereiche. Durch die Natur der rein virtuellen Umgebung sind dies Anwendungen, bei denen aus verschiedenen Gründen keine realen Objekte existieren. Ein wichtiger Bereich stellt hier das Arbeiten mit virtuellen Prototypen im industriellen Umfeld dar. Dies ermöglicht den weitgehenden Verzicht auf die teure und zeitaufwändige Erstellung realer Prototypen. Im Edutainmentbereich besteht das Potential von VR darin Welten und Orte erlebbar zu machen die in dieser Form gar nicht existieren oder aus verschiedenen Gründen so von Menschen nicht erlebt werden können. Beispiele hierfür sind zerstörte, antike Kulturgüter oder der Weltraum.

Das Potential von Augmented Reality liegt dagegen darin, reale Umgebungen für einen Benutzer anders erfahrbar zu machen. Obwohl die reale Welt weiterhin wahrgenommen wird, können zusätzlich Informationen präsentiert werden, die sonst nicht sichtbar sind.

Dadurch, dass der Benutzer sich nicht in eine spezielle Umgebung begeben muss, um die erweiterte Realität erfahren zu können, sind alle Tätigkeiten in der realen Umgebung für AR Unterstützung interessant die erhöhten Erklärungsbedarf besitzen oder die wesentlich attraktiver gestaltet werden können, wenn zusätzliche virtuelle Informationen dazu präsentiert werden. In Bezug auf die oben genannten Beispiele für VR Anwendungen bedeutet dies, dass AR im industriellen Umfeld überall dort eingesetzt werden kann, wo bereits reale Objekte vorliegen, die erweitert oder ergänzt werden sollen. Die spezifischen Anwendungen entstammen dabei nicht nur der Produktentstehung, sondern vielmehr dem kompletten Produktlebenszyklus. Da die Anwendungen von VR und AR Technologien im industriell-

len Umfeld sich grundsätzlich unterscheiden, kann AR als Ergänzung zu bestehenden VR Lösungen betrachtet werden, wodurch die Weiternutzung von virtuellen Informationen, die während der Produktentstehung erzeugt wurden, intensiviert werden kann.

Auch im Bezug auf das Erleben von Kulturgütern unterscheiden sich die Anwendungen von VR und AR grundsätzlich. Während mit VR Orte erlebt werden können, die nicht öffentlich zugänglich sind, kann AR verwendet werden, um reale Orte in anderer Form zu erleben. Ruinen zerfallener Kulturgüter können mit virtuellen 3D Rekonstruktionen überlagert werden, die dem Besucher einen realistischen Eindruck des gleichen Ortes in einem anderen Zeitalter vermitteln. Auch hier können viele Daten gleichermaßen für VR und AR Anwendungen verwendet werden.

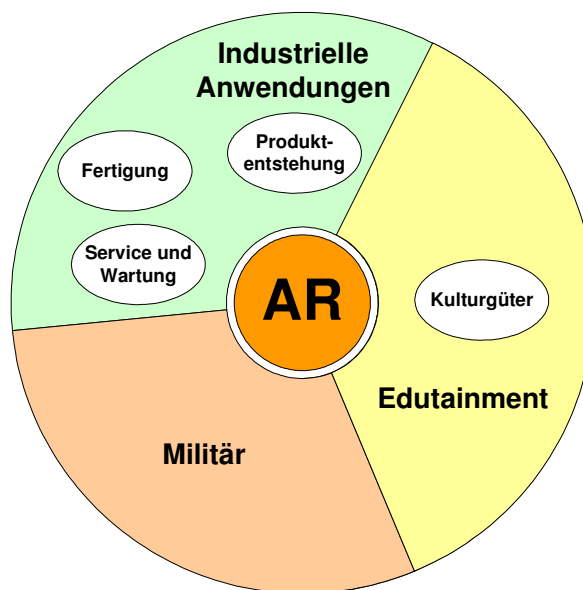


Abbildung 12: Anwendungsbereiche von AR

Neben dieser Verwandtschaft von VR und AR Anwendungen auf den beschriebenen Anwendungsgebieten unterscheiden sich die Anwendungen selbst aber grundlegend. Im Folgenden werden deshalb exemplarische Anwendungen dieser Gebiete genauer betrachtet. Ziel dieser Analyse ist die Identifizierung von Tätigkeiten, die durch AR Unterstützung effizienter oder attraktiver gestaltet werden können.

Nur der Vollständigkeit halber sei darauf hingewiesen, dass ein weiterer wichtiger Bereich für den Fortschritt bei der Entwicklung von AR Technologien der militärische Einsatz zur Unterstützung von Truppen in fremden Umgebungen ist [Livi02]. Im Rahmen dieser Arbeit finden Anwendungen dieses Gebiets jedoch keine Berücksichtigung.

Schwerpunktmäßig werden hier zuerst die Anwendungsgebiete des industriellen Einsatzes von AR betrachtet, in der Reihenfolge, wie sie auch im Produktlebenszyklus vorkommen also beginnend mit der Produktentstehung über die Fertigung bis zu Service und Wartung.

3.1.1 Produktentwicklung

In einigen Bereichen, wie dem Automobilbau, hat sich Virtual Reality als nützliches Werkzeug etabliert. Genau in diesen Bereichen liegen auch detaillierte virtuelle Modelle vor, große Teile der eigent-

lichen Produktentwicklung geschehen bereits Rechnergestützt. VR wird in diesem Prozess beispielsweise für die Beurteilung des Produktdesigns eingesetzt. Anstatt ein reales Modell anzufertigen, wird ein virtuelles Modell auf einer Großprojektion betrachtet und diskutiert. Wie bereits vorher angesprochen, macht der Einsatz von AR erst Sinn, wenn die computergenerierten Daten in Relation zu einem realen Objekt gebracht werden können. Dies setzt voraus, dass bereits ein realer Fahrzeugprototyp vorhanden ist, der durch virtuelle Bauteilvarianten ergänzt oder verändert werden kann.

Von besonderer Bedeutung ist beim Einsatz von Computergraphik in diesem Bereich, unabhängig, ob es sich um VR oder AR handelt, die Qualität der Darstellung. *Visueller Realismus* ist hier mit allerhöchster Priorität zu behandeln, die Anforderung an Tracking und Präsentation sind extrem hoch.

Ein weiteres wichtiges Kriterium für eine AR Lösung zur Unterstützung der Beurteilung von Produktdesigns ist die Unterstützung *mehrerer Benutzer am gleichen Ort*, die diesen Vorgang zusammen durchführen. Den Anwendern muss es Möglich sein, virtuelle *Annotationen* am Modell *anzubringen*, die sie bei der Diskussion des Designs unterstützen.

Mobilitäts- und Ergonomiekriterien besitzen hier bedeutend weniger Gewicht, dürfen aber keinesfalls vernachlässigt werden.

3.1.2 Fertigung

Bei der Fertigung komplexer, technischer Güter kann grob zwischen zwei Kategorien von Tätigkeiten unterschieden werden. Dies sind auf der einen Seite einfache, routinemäßig durchgeführte Tätigkeiten, die oft in gleicher oder sehr ähnlicher Form hintereinander ausgeführt werden. Der Werker, der hier meist nur einen sehr begrenzten Teil des Gesamtfertigungsprozesses durchführt, benötigt in diesem Fall nur bei der Änderung des Fertigungsschritts eine Anleitung, wie er zukünftig diesen Arbeitsschritt durchzuführen hat.

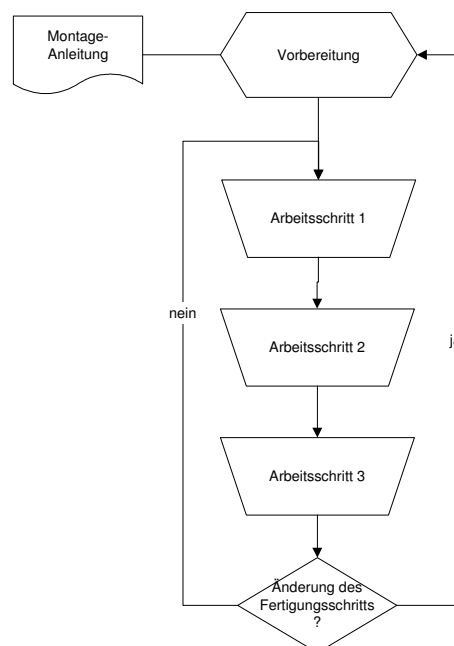


Abbildung 13: Routinemäßig durchgeführte Fertigungstätigkeit

Eine Unterstützung solcher Tätigkeiten mittels AR Technologien erscheint vor allem während des Erlernens des Fertigungsschritts sinnvoll. Während der Durchführung der routinemäßig durchgeführten Arbeitsschritte ist eine AR Unterstützung dagegen nicht notwendig.

Sehr oft treten aber auch bei der Fertigung in Großserien Fertigungsschritte auf, die sich aufgrund von großer Variantenvielfalt oft ändern. Hier kann auch während der Fertigung ein AR System wichtige Hilfestellungen geben. So kann beispielsweise bei der Fertigung von unterschiedlichen Varianten automatisch die aktuell zu bearbeitende Variante angezeigt und bei seltenen Varianten auch eine weitergehende Unterstützung geboten werden.

Führt der Werker selbst komplexere Tätigkeiten durch, die auch noch oft und stark variieren, wie es für Kleinserienfertigung typisch ist, benötigt er auch mehr Einblick in die Dokumentation der Montagetätigkeit. Da sich bei solchen Tätigkeiten weniger routinemäßiges Verhalten einstellen kann, steigt auch der zeitliche Aufwand des Studiums der Dokumentation. Die richtige Dokumentation muss vorliegen und die aktuelle Position innerhalb der Dokumentation angesteuert werden. Besonders wenn die Dokumentation in Papierform vorliegt, kann dies sehr zeitintensiv sein und verlangt zumindest die Unterbrechung der aktuellen Tätigkeit. Liegt die Dokumentation in elektronischer Form vor, kann durch effiziente Verlinkung der Dokumentation und Suchfunktionen immerhin der zeitliche Aufwand für die Navigation innerhalb der Dokumentation verkürzt werden. Zusätzlich ist es auch möglich, den Werker basierend auf seinem Arbeitsplan bereits automatisch mit der richtigen Dokumentation zu versehen. Werden die Informationen dann in einem AR System präsentiert, ist auch das Unterbrechen der Tätigkeit zwecks Zugriffs auf die Dokumentation nicht mehr nötig.

Zusätzlich zur Erleichterung und Beschleunigung der Tätigkeiten des Werkers, kann AR auch als Werkzeug zur Qualitätssicherung verwendet werden. Dies betrifft sowohl die AR Unterstützung des Werkers als auch eine Anwendung von AR Technologien in der Qualitätssicherung. Der Anwender muss die korrekte Durchführung kritischer Arbeitsschritte quittieren, und wird somit wiederholt direkt seiner Mitverantwortlichkeit an der Produktqualität erinnert. Bei der Qualitätssicherung selbst kann eine Abnahmeanleitung präsentiert werden, die durch den Prüfprozess leitet. Bei einer entsprechenden Anbindung des AR Systems an das technische Produkt kann es auch zur kontextbasierten Darstellung von Testwerten und Zustandsdaten verwendet werden. Der Benutzer kann vor einer laufenden Anlage die Prozessdaten verschiedener Teilaggregate einsehen und so einen Überblick über die korrekte Funktion der Anlage erhalten.

Bei der Herstellung komplexer, technischer Produkte, wie Automobilen und Flugzeugen kann die Unterstützung durch Augmented Reality in zwei Klassen eingeteilt werden. Zum einen kann AR zum Training von Arbeitsabläufen verwendet werden, zum anderen bei der Unterstützung während der Montage selbst. Wiederholt sich die Tätigkeit eines Monteurs oft, benötigt er meist eine einmalige Unterweisung in die neue Tätigkeit. Typischerweise besitzt die zu trainierende Tätigkeit überschaubare Komplexität. Der Arbeiter bekommt die neu zu erlernende Tätigkeit Schritt für Schritt in seinem Blickfeld präsentiert. Ist der Arbeitsablauf erlernt, ist die AR Unterstützung im Allgemeinen nicht mehr notwendig. Reiners et al [REIN98] beschreibt ein solches Trainingssystem, das beispielhaft die Montage eines Türschlosses in eine Autotür erklärt.

Bei komplexeren oder oft wechselnden Tätigkeiten, sei es durch eine große Zahl an Varianten des zu fertigenden Produkts, bietet sich eine Unterstützung während der Durchführung an. Es können so unterstützende Informationen zum jeweiligen Arbeitsschritt gegeben werden, die im Idealfall speziell auf den Anwender abgestimmt sind. Ein Beispiel für ein solches System ist das von Caudell und Mizell präsentierte System zur Fertigung von Kabelbäumen für den Flugzeugbau [CAU92].

Die Ziele der beiden Unterstützungen können dabei sehr unterschiedlich sein. Während beim Training einer Tätigkeit im Vordergrund steht, dass der Anwender in die Lage versetzt wird, die Tätigkeit zukünftig selbstständig durchzuführen, ist das Ziel bei der Unterstützung während der Montagetätigkeit selbst, den Monteur möglichst effizient durch die unterschiedlichen Arbeitsschritte zu leiten. Ein Lerneffekt steht hierbei nicht im Vordergrund. Es ist allerdings auch die Kombination beider Unterstützungsformen denkbar.

Von der Häufigkeit und Dauer der Anwendung her betrachtet, ist im Trainingsfall davon auszugehen, dass ein Monteur nur beim Wechsel seiner Tätigkeit in die neuen Arbeitsschritte eingeführt werden muss. Die Unterstützung während der Montage erfordert dagegen, dass der Anwender das AR System bei seiner Tätigkeit immer mit sich führt, so dass die Assistenzfunktion bei Bedarf sofort zur Verfügung steht.

Der mit AR Assistenzfunktionalität abzudeckende Arbeitsbereich ist bei den unterschiedlichen Produkten Flugzeug und Automobil naturgemäß von sehr unterschiedlicher Größe und Gestalt. Die Extreme stellen hier Flugzeugrumpf und Motorraum des Autos dar. In jedem Fall muss die Tätigkeit des Monteurs aber als mobil bezeichnet werden.

Aus Sicht der AR Basisfunktionalität können beide Szenarien gleich behandelt werden. Die Tätigkeit des Anwenders muss im Rechner so modelliert sein, dass in jedem Arbeitsschritt genau die relevanten Informationen verfügbar stehen. Das AR System wird so in die Lage versetzt, den Benutzer Schritt für Schritt durch seine Tätigkeit zu leiten. Im Folgenden nennen wir diese Art der AR Unterstützung auch *geführter Modus*.

Da das Sichtfeld der meisten Displays, wie sie für Augmented Reality Anwendungen verfügbar sind, relativ klein ist, kann es schnell passieren, dass Objekte, die für den aktuellen Arbeitsschritt wichtig sind, nicht im Blickfeld liegen. Das AR System soll in solchen Fällen den Blick des Benutzers zu diesem Ort dirigieren. Diese Unterstützung wird nachfolgend *Nahbereichsnavigation* genannt.

Besonders bei der Unterstützung während der Tätigkeit über einen ganzen Arbeitstag ist die Ergonomie von mobilen AR Systemen besonders von Bedeutung. Das System darf im wahrsten Sinne des Wortes nicht zur Last fallen. Dies gilt im Besonderen für die Bedienung des AR Systems. Da der Benutzer bei der Durchführung seiner realen Aufgabe unterstützt werden soll, darf die Bedienung des Hilfswerkzeugs nicht von der realen Arbeit ablenken oder bei dieser behindern. Ein weiterer wichtiger Aspekt in diesem Umfeld ist die Robustheit sowohl von Hard- als auch Software. Ausfallzeiten eines solchen Assistenzsystems sind im Fall der Unterstützung während der Montage fast gleichbedeutend mit dem Ausfall des Arbeitsplatzes. Es ist also auch von einem nicht unerheblichen Aufwand bei der Unterstützung des Einsatzes von AR Assistenzfunktionen zu rechnen. Speziell wenn man von einer großen Anzahl von AR Systemen in der Fertigung von Automobilen oder Flugzeugen ausgeht müssen IT Servicepersonal und Backupsysteme zur sofortigen Verfügung stehen.

Ein gesondertes Problem stellt hierbei die Erstellung von Arbeitsanweisungen für Augmented Reality Assistenzfunktionalitäten dar. Auch wenn bereits ein Produktdatenmodell (PDM) und Geometriemodelle vorliegen, ist die Erstellung von AR gestützten Anleitungen mit großem Aufwand verbunden und erfordert die Bereitstellung spezialisierter Autorenwerkzeuge. Zauner et al [ZAU03] präsentieren ein MR System zur Unterstützung der Montage von hierarchischen Strukturen, das die Erstellung der Anweisungen in einer MR Umgebung unterstützt.

3.1.3 Service und Wartung

Service und Wartung bilden das wohl klassischste Anwendungsgebiet für Augmented Reality Unterstützung. Bei der überwiegenden Zahl an Tätigkeiten im Serviceumfeld spielt der *geführte Modus* eine wichtige Rolle. Der Ablauf vieler Servicetätigkeiten lässt sich als die Hintereinanderausführung verschiedener Arbeitsschritte beschreiben. Besonders bei Wartungs- und Instandhaltungsarbeiten liegt oft ein fester Arbeitsplan vor.

Handelt es sich bei der Servicetätigkeit allerdings um die Behebung eines Fehlerfalls an einer technischen Anlage, sieht das Vorgehen im Allgemeinen grundlegend anders aus. Je nach Qualität der Fehlerbeschreibung muss erst das Fehlerbild analysiert und die Ursache ermittelt werden. Hierfür ist im Allgemeinen die Verfügbarkeit einer wesentlich größeren Menge an Daten notwendig, als im *geführten Modus*. Da der Benutzer frei durch den verfügbaren Datenbestand navigieren kann, sprechen wir in diesem Zusammenhang auch vom *freien Modus*. Typischerweise kann es bei der Durchführung einer Tätigkeit zum Wechsel zwischen *freiem* und *geführtem* Modus kommen. Tritt beispielsweise beim Abarbeiten einer Aufgabenbeschreibung ein unvorhergesehenes Ereignis auf, muss der Benutzer in die Lage versetzt werden, den Arbeitsablauf zu unterbrechen und zu versuchen im freien Modus hilfreiche Informationen zu erhalten, die ihn in dieser Situation weiterhelfen. Ist das Problem behoben, kann die Arbeit wieder im aufgabenbasierten Modus fortgesetzt werden. Andersherum kann sich auch unter den Informationen, die der Benutzer im freien Modus durchstöbert eine Arbeitsbeschreibung befinden, die das System dazu veranlasst, in den *geführten* Modus zu wechseln. Ein Beispiel für einen solchen Wechsel ist der Übergang von der Fehleranalyse zur Behebung.

Neben dem Wechsel aus dem *geführten* in den *freien Modus*, um auf eine größere Informationsmenge zugreifen zu können, kann die Notwendigkeit bestehen, dass ein zusätzlicher Experte hinzugezogen werden muss. Dies geschieht klassischerweise über einen per Telefon zugeschalteten *Remote-Experten* oder im besten Fall eine Bildübertragung. Oft ist dabei die Kommunikation über Telefon nicht ausreichend, was zur Konsequenz hat, dass ein weiterer Techniker zur Unterstützung geschickt werden muss. Ein *Remote-Experte* mit Augmented Reality Unterstützung bietet hier erheblich weitergehende Möglichkeiten der Kommunikation. Der Techniker vor Ort und der *Remote-Experte* teilen die erweiterte Sicht des Technikers, wobei der *Remote-Experte* Annotationen im Blickfeld des Technikers direkt an der Maschine anbringen kann.

Während der Anwender in der Produktion meist bestens Vertraut mit seinem Arbeitsumfeld ist, ist hiervon speziell bei der Wartung von großen Anlagen nicht auszugehen. Besonders an fremden, weitläufigen Orten kommt die Unterstützung bei der *Navigation* hinzu. Der Anwender wird dabei vom AR System an den eigentlichen Einsatzort geleitet.

Vielfach ist diese Art der *Navigation* nicht wirklich notwendig, beispielsweise wenn es sich um den Automobil-Service handelt. Der Techniker steht üblicherweise bereits vor dem betreffenden Fahrzeug. Allerdings ist auch hier ebenfalls der Einsatz von Techniken zur Unterstützung der *Nahbereichsnavigation* sinnvoll. Der Unterschied zwischen *Navigation* und *Nahbereichsnavigation* ist, dass die *Navigation* eine separate Aufgabe des Anwenders darstellt, während die *Nahbereichsnavigation* als Hilfsmittel während der Durchführung einer anderen Tätigkeit zu sehen ist. Folgt der Benutzer den Anweisungen der *Navigationsunterstützung*, muss er einen kompletten Ortswechsel vornehmen. Die *Nahbereichsnavigation* erfordert im Allgemeinen nur eine Änderung der Blickrichtung.

Eine besondere Anforderung stellt die Berücksichtigung von *Veränderungen im Arbeitsraum* dar. Neben Händen, Armen und Werkzeugen, die Teile des Sichtfeldes zeitweilig verdecken, ist hierunter das Entfernen und Anbringen vor allem großer Bauteile zu verstehen. Das Entfernen einer großen Abdeckung hat sowohl auf das Tracking als auch auf die Präsentation starken Einfluss. Das virtuelle Modell muss den neuen realen Gegebenheiten nachgeführt werden. Das Tracking-Verfahren muss dabei weiterhin die Position und Blickrichtung des Anwenders erfassen und zusätzlich das Entfernen eines großen Bauteils verfolgen. Im Idealfall sollten solche Änderungen am realen Objekt verwendet werden können, die Ablaufsteuerung des AR Systems anzusteuern. Auch Prozessdaten, die über eine Schnittstelle aus einer Anlage gelesen werden können, sollten zur Steuerung verwendet werden können. Die Interaktion mit einem solchen System wird somit weiter vereinfacht.

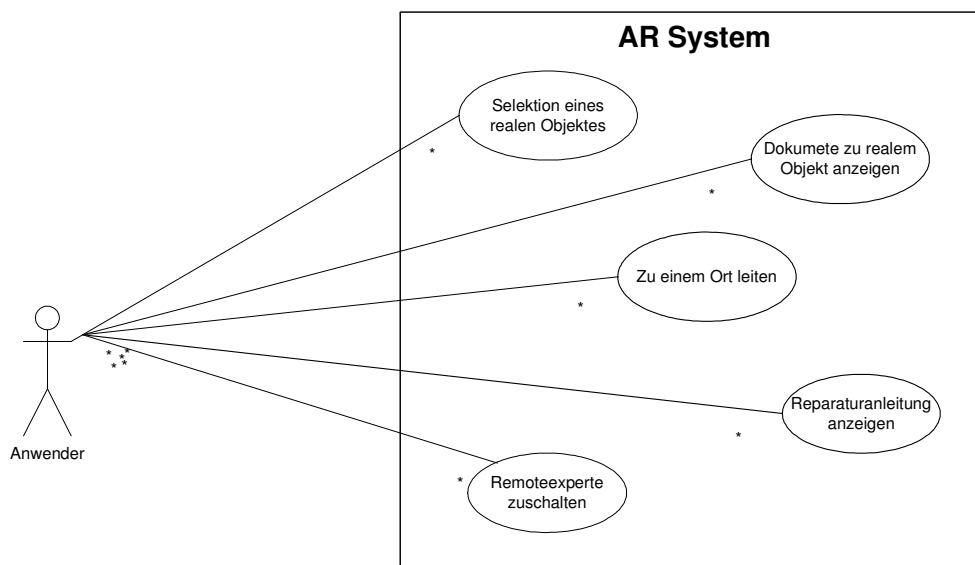


Abbildung 14: Use-Case: Service

3.1.4 Kulturelles Erbe

Neben dem industriellen Bereich, bietet vor allem der private Bereich eine große Menge potentieller Anwendungsgebiete für AR Anwendungen. Stellvertretend sind hier die Bereiche Tourismus und kulturelles Erbe zu nennen. Der Augmented Reality Guide stellt die logische Erweiterung des Audio Guides dar. Besucher von archäologischen Stätten oder von Museen sind vertraut mit Audio Guides, die sie beim Besuch leiten und mit kontextabhängigen, meist gesprochenen Informationen versorgen. Ein

Augmented Reality Guide muss gegenüber solchen Systemen einen entsprechenden Mehrwert für den Benutzer bringen.

Neben Texten, Standbildern und Videos, die kontextabhängig präsentiert werden, können auch mit der realen Umgebung registrierte 3D Szenen überlagert werden. Die Präsentation der Rekonstruktion eines Tempels, der heute nur noch als Ruine vorhanden ist, kann dem Benutzer einen Eindruck vermitteln, wie die Stätte in der Antike aussah. Anhand belebter, animierter 3D Szenen kann die Lebensweise der damaligen Bewohner erfahren werden.

Die Interaktion mit einem solchen System muss so einfach und klar verständlich sein, dass auch Besucher, die nicht mit der Bedienung von Computern vertraut sind, damit umgehen können. Das System muss quasi ohne Einarbeitungszeit durch Laien verwendbar sein, die Bedienung muss zu diesem Zweck vollständig selbsterklärend sein. Zusätzlich ist an allen bedeutenderen Grabungsstätten mit internationalem Publikum zu rechnen, so dass ein System multilingual und wo dies möglich ist, besser noch sprachneutral zu gestalten ist. Auch die Verwendung durch Besucher unterschiedlichsten Alters muss bei Interaktion und Präsentation der Inhalte berücksichtigt werden.

Typischerweise werden solche Systeme gleichermaßen für die Nutzung durch Besucher einer antiken Ausgrabungsstätte vorgesehen, wie für Archäologen, die allerdings vollständig andere Inhalte benötigen und das System auch anders nutzen.

Eine besondere Rolle spielt in diesem Zusammenhang das Engineering der Anwendung und dabei vor allem das Erstellen der Anwendungsdaten. Archäologen sollten vom System in die Lage versetzt werden, selbst Inhalte zu generieren, also die virtuellen Beschreibungen von Artefakten einzubringen und korrekt zu platzieren, ohne sich vorher tiefer gehende Kenntnisse über zugrunde liegende Technologien wie Datenbanken oder 3D Computergrafik aneignen zu müssen.

3.2 Anwendergruppen

Neben den unterschiedlichen Gebieten, in denen ein Endanwender AR Unterstützung für die Durchführung anwendungsspezifischer Aufgaben erhält, sind auch die verschiedenen Anwendergruppen die mit einem AR System in Kontakt kommen und deren Anforderungen an ein solches System zu betrachten. Hierbei handelt es sich vor allem um die Endanwender, die Applikationsentwickler und Autoren, die Inhalte für Anwendungen generieren und Systementwickler, die Erweiterungen für das Basis-System erstellen.

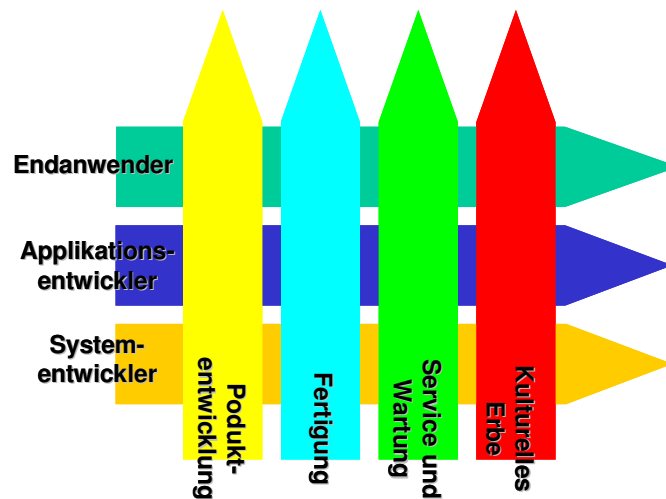


Abbildung 15: Benutzergruppen und Anwendungsgebiete

3.2.1 Endanwender

Im Mittelpunkt bei der Realisierung einer neuen Mensch-Computer Schnittstelle steht der Anwender mit seinen Anforderungen und Bedürfnissen, weshalb man in diesem Zusammenhang auch von benutzerzentrierter Systemgestaltung spricht.

Unabhängig von der spezifischen Aufgabe des Benutzers verspricht sich dieser typischerweise Unterstützung bei einer realen Tätigkeit. Augmented Reality verspricht hier allein durch die Möglichkeit Informationen lagerichtig im Blickfeld des Benutzers präsentieren zu können, einen Gewinn. Das allein reicht für den realen Einsatz aber bei weitem nicht aus. Besonders die Qualität der Informationen und die Form der Präsentation tragen entscheidend zum erfolgreichen Einsatz bei. Abgeleitet aus der Arbeitssituation des Anwenders und dessen Blickrichtung sollen die aktuell relevanten Informationen in einer Form präsentiert werden, in der diese leicht erfasst und verarbeitet werden können. Die Bedienung des Systems sollte ihn nicht so sehr in Anspruch nehmen, dass er von seiner eigentlichen Aufgabe abgelenkt wird. Außerdem ist es wichtig, dass die Bedienung des Systems leicht erlernbar ist und der Bedarf für Schulungen zum Umgang mit dem System minimal ist.

Neben diesen allgemeinen Anforderungen haben die Benutzer je nach Art ihrer Tätigkeit, spezielle Anforderungen an ein AR System und die Anwendung, die darauf basiert. Diese speziellen Anforderungen einiger Anwendergruppen bilden den Hauptbestandteil von Abschnitt 3.1 und werden deshalb hier nicht wiederholt.

3.2.2 Applikationsentwickler

Basierend auf den AR Kernfunktionalitäten entwickelt der Applikationsentwickler die eigentliche Anwendung. Diese Aufgabe besteht aus der Konfigurierung des Systems, der Erstellung der Applikationslogik, dem Zusammenstellen und Aufbereiten der Daten, die dem Anwender präsentiert werden und der Bereitstellung angemessener Interaktionsmechanismen. Hauptanforderung ist hier, dass diese Aufgaben mit möglichst minimalem Aufwand erledigt werden können, da vor allem in diesem Bereich großer Aufwand entstehen kann, die dem Nutzen beim Einsatz des Systems gegenübergestellt werden

müssen. Die Crux liegt hierbei darin, dass der Einsatz von Augmented Reality basierten Assistenzfunktionen besonders dort sinnvoll ist, wo komplexe Tätigkeiten zu unterstützen sind, die nicht häufig von einem Anwender durchgeführt werden. Der Benutzer ist in solchen Fällen besonders auf qualitativ hochwertige Unterstützung angewiesen, da er mit der Tätigkeit nicht vertraut ist. Die Assistenzfunktionalität muss also mit besonders großer Sorgfalt erstellt werden, was im Allgemeinen großen Aufwand bedeutet. Demgegenüber steht eine seltene Nutzung, die eine Amortisierung des Erstellungsaufwands erschwert. Um dem Applikationsentwickler bestmöglich bei seiner Tätigkeit zu unterstützen, müssen Schnittstellen und Werkzeuge vorhanden sein, die möglichst effiziente und an die Bedürfnisse technischer Redakteure angepasste Funktionalitäten bieten um bestehende Daten zu übernehmen oder neue zu generieren. Teile bestehender AR Anwendungen sollten wieder verwendbar sein, so dass der Aufwand zukünftiger Entwicklungsaufgaben weiter reduziert werden kann.

Da eine Hauptaufgabe bei der Erstellung interaktiver Anwendungen die Gestaltung von Benutzerschnittstellen ist, sollten die Elemente der Benutzerschnittstelle und die Interaktionstechniken in hohem Maße wieder verwendbar und konfigurierbar sein, um den hierbei entstehenden Aufwand zu reduzieren.

3.2.3 Systementwickler

Die Systementwickler legen mit der Erstellung des Frameworks und den darin enthaltenen Basisfunktionalitäten die Grundlage für die Anwendungen, die darin realisiert werden. Ihre Aufgabe ist ebenso, applikationsspezifische Systemerweiterungen zu realisieren, wie die Implementierung zusätzlicher oder verbesserter Systemfunktionalitäten.

3.3 Analyseergebnisse

3.3.1 Arbeitsmodi

Wie in Abschnitt 3.1.2 dargelegt, kann grundsätzlich zwischen zwei Modi der Unterstützung durch ein Augmented Reality System unterschieden werden. Zum einem den **geführten Modus**, bei dem die Aufgabe, die ein Anwender durchführen soll, bereits im Voraus bekannt und im Rechner modelliert ist. Eine solche Aufgabe kann durch die sukzessive Ausführung von Arbeitsschritten beschrieben werden. Das AR-System gibt die Anweisungen zu den einzelnen Schritten nacheinander wieder und leitet den Benutzer so Schritt für Schritt durch seine Tätigkeit. Mit jedem Arbeitsschritt sind Informationen verknüpft, die genau dann zur Verfügung stehen, wenn der Benutzer sich in diesem Arbeitsschritt befindet.

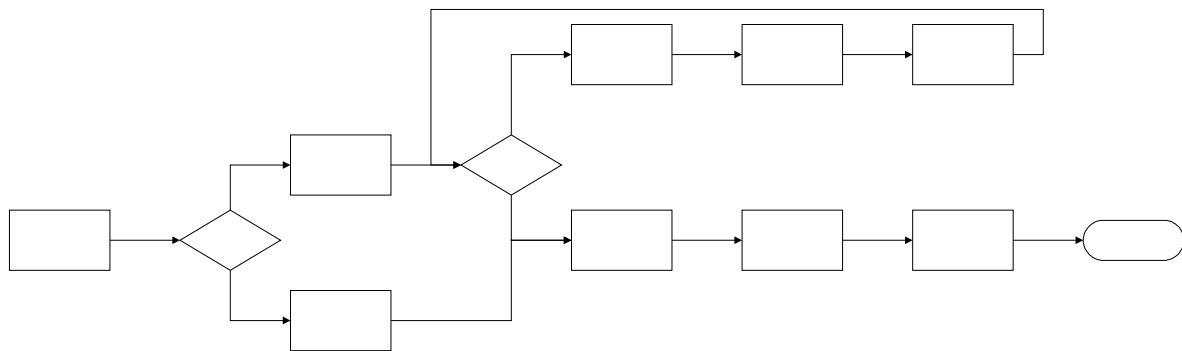


Abbildung 16: Beispiel-Arbeitsablauf

Der zweite Modus kann als **freier Modus** bezeichnet werden. Der Benutzer ist nicht an eine präzise Aufgabenbeschreibung gebunden, sondern kann wahlfrei auf Informationen zugreifen, die mit den Objekten der realen Umgebung verknüpft sind. Zu berücksichtigen ist, dass es während einer Anwendung durchaus zu Wechseln zwischen den beiden Modi kommen kann.

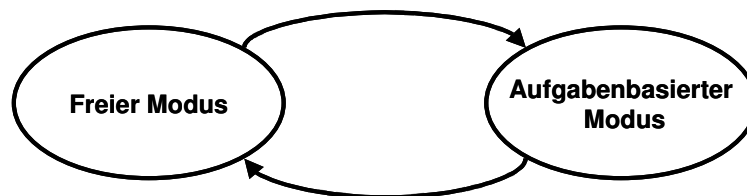


Abbildung 17: Moduswechsel

In der Beschreibung der Anwendungsbereiche wurden bereits einige Szenarien für solche Modiwechsel vorgestellt. Charakteristischerweise kann der Wechsel von einem Modus in den anderen meist nicht vorhergesagt werden.

Bei der Unterstützung der unterschiedlichen Arbeitsmodi in einem AR System sind verschiedene Aspekte zu betrachten. Zuerst die unterschiedliche Art der Präsentation von Informationen, die in beiden Modi erforderlich ist, die Art, wie mit einem AR System in den verschiedenen Modi interagiert wird und vor allem auch, wie die Daten organisiert sein müssen, um beide Modi optimal unterstützen zu können.

Im geführten Modus steht bereits im Vorhinein fest, welche Daten dargestellt werden sollen. Die Präsentation kann also sehr gut bereits während der Erstellung der Arbeitsschrittbeschreibung gesteuert werden. Dies beinhaltet vor allem eine Reduzierung auf die wesentlichen Informationen, die zur Durchführung eines Arbeitsschrittes notwendig sind.

Die Darstellung von Informationen im freien Modus dagegen ist wesentlich schwieriger zu kontrollieren, da hier nicht präzise bestimmt werden kann, welche Daten aktuell für den Benutzer von Relevanz sind. Die Aufgabe besteht also darin, aus der aktuellen Situation, in der sich der Anwender befindet, Rückschlüsse zu ziehen, welche Informationen von Interesse sein könnten und welche nicht. Die Informationen über die Situation des Anwenders, also dessen aktuelle Aufgabe, den Ort, an dem er sich befindet, die Zeit und persönliche Informationen, die den Anwender beschreiben, bilden den Kontext, der dem AR System zur Verfügung steht. Basierend auf dem **Kontext** kann nun eine **Filterung** der zu präsentierenden Daten erfolgen. Ziel der Filterung ist in erster Linie eine Informationsüberflutung des

Anwenders zu verhindern. Den zur Verfügung stehenden Informationsraum gilt es möglichst effektiv einzugrenzen. Neben der reinen Präsentation der Informationen dient die Informations-Filterung auch der Reduzierung der Datenmenge die zu einem mobilen Endgerät transportiert werden muss. Neben der begrenzten Bandbreite, die vor allem in drahtlosen Übertragungsmedien zur Verfügung steht, dient dies auch der Reduktion von Wartezeiten und nicht zuletzt der Kostenreduzierung. Da es, wie bereits beschrieben, nicht möglich ist, die benötigten Informationen im Voraus zu bestimmen, müssen also vor allem umfangreiche Datensätze durch entsprechend reduzierte Versionen repräsentiert werden. Diese **Preview**-Präsentationen sollten im Idealfall automatisch erzeugt werden, um an dieser Stelle keinen zusätzlichen Authoring-Aufwand entstehen zu lassen.

Die Interaktionsaufgabe im geführten Modus besteht vor allem in der Bestätigung bereits erledigter Arbeitsschritte und der Wahl des nächsten Schrittes im Ablauf. Zusätzlich werden noch die Möglichkeit der Wiederholung eines Schrittes und die Auswahl einer Alternative benötigt, für den Fall, dass die Beschreibung eines Arbeitsablaufs Verzweigungen aufweist.

Im freien Modus dagegen besteht die hauptsächliche Interaktionsaufgabe des Anwenders darin, ein Datenobjekt auszuwählen und über dieses Objekt weitere Informationen zu einem realen Objekt zu erhalten. Je nach Größe und Dichte des Informationsraums und auch abhängig von der jeweiligen Anwendung sind hier unterschiedliche Selektionsmechanismen vorzusehen.

Die zu präsentierenden Daten im geführten Modus bestehen im Wesentlichen aus den einzelnen Stationen oder Arbeitsschritten, die nacheinander besucht bzw. abgearbeitet werden. Einer dieser Schritte beinhaltet dabei typischerweise ein oder mehrere Informationsobjekte. Datenobjekte werden in einem Arbeitsschritt sichtbar, wenn sie voraussichtlich benötigt werden. Eine spezielle Organisation der Daten ist hier nicht notwendig.

Im freien Modus dagegen werden die Daten über ihre zugehörige räumliche Struktur adressiert. Der Bezug zwischen einem Datenobjekt und einem Ort oder einem Objekt der realen Umgebung muss hergestellt werden. Zusätzlich sind Mechanismen zu realisieren, die auf Ebene der Datenorganisation die Filterung der Informationen unterstützen. Metainformationen unterstützen die Auswahl der Dokumente, die zu einem bestimmten Anwender- oder Aufgabenprofil passen. Räumliche Informationen, wie die Gruppierung in Räumen eines Gebäudes oder anderen Strukturen erlauben die gezielte Steuerung der Sichtbarkeit von Objekten.

3.3.2 Navigationsunterstützung

Die Aufgabe der Navigationsunterstützung besteht darin, den Anwender von seiner aktuellen Ist-Position zu einer Zielposition zu leiten. Bei der Zielposition kann es sich sowohl um eine zu wartende Maschine in einer Werkshalle als auch den nächsten Wegpunkt bei der Tour über eine archäologische Ausgrabungsstätte handeln. Zu unterscheiden ist dabei zwischen einer im Voraus festgelegten, **statischen Route** und einer automatisch generierten, sich an neue Gegebenheiten **adaptierende Route**. Während im ersten Fall die Route bereits während eines Authoring-Prozesses festgelegt wird, ist bei der sich anpassenden Route eine zusätzliche Laufzeit-Komponente notwendig, die kontinuierlich Ist- und Sollwerte vergleicht und basierend auf Abweichungen Anpassungen an der Route vornimmt.

Für den zweidimensionalen Fall der Navigationsunterstützung liegen bereits viele Erkenntnisse aus dem Bereich der Navigationssysteme für Automobile vor. Auch diese Systeme sollen den Fahrer bei der Findung des Weges zu einem vorher bestimmten Ziel unterstützen. Eine der wichtigsten Anforderungen an ein System in diesem Bereich ist, dass der Fahrer möglichst wenig von seiner eigentlichen Aufgabe, dem Führen eines Automobils abgelenkt wird.

Die überwiegende Anzahl der Navigationsaufgaben kann als 2D Problem beschrieben werden, da der Anwender sich typischerweise nur horizontal bewegen kann. Es ist also ausreichend, wenn hinreichend genaue 2D Informationen über die Ist-Position und die aktuelle Blickrichtung des AR Systems vorliegen. Ein Tracking-System, dass für AR-Anwendungen tauglich ist, übertrifft diese Anforderungen deutlich. Es ist allerdings zu gewährleisten, dass die Tracking-Informationen auch flächendeckend entlang aller potentiellen Wegstrecken verfügbar sind.

Ein Sonderfall stellt die Navigation in mehrstöckigen Gebäuden dar. Hier ist die 2D Tracking-Information nicht eindeutig und somit nicht ausreichend. Aber auch hier ist keine „echte“ 3D Information notwendig, da sich der Anwender nur auf Stockwerkebene bewegen kann. Es handelt sich also eher um ein 2½D Problem. Zur Auflösung der Mehrdeutigkeit ist lediglich die Angabe des aktuellen Stockwerks notwendig. Innerhalb eines Stockwerks genügt wieder die 2D Navigation. Da auch ein Wechsel des Stockwerks nur an speziellen Übergängen, Treppenhäusern oder Aufzügen, vorgenommen werden kann, ist lediglich an diesen Punkten eine Sonderbehandlung notwendig.

3.3.3 Nahbereichsnavigation

Ein Problem von verfügbaren Displaysystemen, wie sie in AR Systemen verwendet werden, ist das eng begrenzte Blickfeld. Dies bedingt, dass ein Teil der Informationen außerhalb des aktuell sichtbaren Bereichs der Szene liegt. Im geführten Modus kann dies leicht dazu führen, dass die Informationen für den nächsten durchzuführenden Schritt nicht sichtbar sind, da sich der Benutzer zwar am richtigen Ort befindet, aber in eine andere Richtung blickt. Es wird also eine Funktionalität benötigt, die seinen Blick in die gewünschte Richtung dirigiert. Obwohl die Informationen räumlich eng beieinander liegen, kann der Winkel aus der Betrachterposition zwischen zwei Informationsobjekten, bedingt durch den geringen Abstand des Anwenders von den Informationsobjekten, beträchtlich sein.

Neben dem Einsatz im geführten Modus besitzt die **Nahbereichsnavigation** auch im freien Modus Bedeutung, wenn ein Informationsobjekt auf ein anderes Objekt verweist, dass sich außerhalb des Blickfelds befindet.

3.3.4 Kontextvisualisierung

Im freien Arbeitsmodus wirkt sich das enge Sichtfeld von AR Displaysystemen dadurch aus, dass nur ein kleiner Ausschnitt des umgebenden Informationsraumes wahrgenommen werden kann. Der im Verhältnis größere Teil des Informationsraums befindet sich außerhalb des Sichtfelds. Da in diesem Modus nur bedingt eingegrenzt werden kann, welche Informationen von Interesse sind, ist es notwendig, dem Anwender einen Überblick über die ihm zur Verfügung stehenden Informationen zu verschaffen. Die **Kontextvisualisierung** hilft hierbei durch die Bereitstellung geeigneter Darstellungsformen für relevante Informationen, die sich außerhalb des Blickfelds des Anwenders befinden.

3.3.5 Informationspräsentation

Die Präsentationsinhalte, die in einem System darzustellen sind, das zum Ziel hat, antikes Leben virtuell nachzustellen, können wesentlich komplexer ausfallen, als dies im industriellen Umfeld typischerweise der Fall ist. Lebensnahe virtuelle Charaktere, die zwischen und in realistischen Rekonstruktionen der antiken Gebäude Szenen des damaligen Lebens nachstellen, stellen sowohl an die Laufzeitumgebung des AR Systems als auch an die Autorenwerkzeuge höchste Anforderungen. Die Darstellung in solchen Anwendungen soll einen realistischen Eindruck der gezeigten virtuellen Inhalte vermitteln. Dazu gehören die realistische Wiedergabe von Materialeigenschaften, die Berücksichtigung der realen Beleuchtungssituation und die korrekte Behandlung von Verdeckungen zwischen virtuellen und realen Objekten.

Bei der Darstellung von Anweisungen zu einer Service-Aufgabe besitzen diese Anforderungen bedeutend weniger Gewicht, visueller Realismus als eigenes Ziel existiert hier nicht. Allerdings dienen Techniken wie Verdeckungsdarstellung auch der Erfassung einer Szene als Ganzes. Die räumliche Lage eines virtuellen Objekts in Relation zur realen Umgebung kann erst durch die Berücksichtigung von Verdeckungen erfasst werden. Ansonsten werden in diesem Umfeld vor allem Techniken benötigt, die maximale Erkennbarkeit von virtuellen Informationen vor der realen Umgebung gewährleisten. Die virtuellen Informationen sollen weniger als Teil einer Gesamtszene wahrgenommen werden, sondern vielmehr direkt die Aufmerksamkeit des Benutzers auf sich ziehen.

3.3.6 Informationsfilterung

Bedingt durch den Einsatz weniger performanter mobiler Hardware und die begrenzten Möglichkeiten der menschlichen Informationsverarbeitung gilt es die Menge der zu präsentierenden Daten bzw. Informationen effizient zu reduzieren. In gut kontrollierten Anwendungen, wie der geführten Service Unterstützung, lässt sich die Menge der Informationen, die zur gleichen Zeit präsentiert werden, bereits bei der Erstellung der Anwendung steuern. In anderen Szenarien, bei denen sich der Anwender frei durch den Informationsraum bewegt, ist die vorherige Steuerung der Menge darzustellender Informationen nicht möglich. Hier müssen Mechanismen greifen, die zur Laufzeit eine Reduktion der darzustellenden Datenmenge vornehmen, so dass Display und in letzter Konsequenz auch der Anwender von einer Informationsüberflutung bewahrt werden.

3.3.7 Interaktion

Bei der Realisierung der **Interaktion** mit einem AR System für die beschriebenen Anwendungsbereiche ist zu berücksichtigen, dass der Anwender hauptsächlich einer anderen Tätigkeit in der realen Welt nachgeht. Die Bedienung des AR Systems stellt also eine zusätzliche Aufgabe für ihn dar. Sie sollte deshalb so einfach wie möglich erfolgen und mit einer minimalen Menge an Eingabegeräten auskommen, die intuitiv bedienbar sind. Vor allem für Benutzer, die nicht mit VR/AR-Interaktionstechniken vertraut sind, sollte sich die Bedienung ohne lange Einarbeitung erschließen. Dies ist besonders dann wichtig, wenn das System nur für kurze Zeit an einen Benutzer gegeben wird, wie zum Beispiel im Szenario der Präsentation von Kulturerbe. Hier ist eine Einarbeitungsphase bei Besuchern, die das System nur kurz benutzen nicht akzeptabel.

3.3.8 Kooperation

Bei der **Kooperation** von Anwendern eines AR Systems sind zwei Fälle zu unterscheiden. Die Anwender befinden sich am gleichen Ort oder an unterschiedlichen Orten. Beispiele für Anwendungen, bei denen mehrere Benutzer eine Mixed Reality Welt teilen sind der Remote Experte mit AR Unterstützung und der Design Review in AR. Ersteres ein Szenario, bei dem sich beide Benutzer an unterschiedlichen Orten befinden, während letzteres ein Beispiel für ein Szenario ist, bei dem sich mehrere Benutzer am gleichen Ort befinden. Besonders in solchen Anwendungen können die Anwender auch weiterhin natürlich miteinander kommunizieren, sowohl verbal als auch non-verbal.

Die Anforderungen, die ein solches Szenario stellt, betreffen vor allem die Interaktion mit den virtuellen Objekten und deren Synchronisation. Jeder Anwender kann dabei gemeinsame und auch private, nur für ihn sichtbare Informationen sehen.

Im Remote Experten Szenario dagegen ist die Kommunikation zwischen den beiden Anwendern auf eine zusätzliche Sprachübertragung angewiesen. Neben dieser bidirektionalen Sprachübertragung ist auch die Video-Übertragung zum Experten notwendig. Es treten also erhebliche Anforderungen an Übertragungs-Bandbreiten auf. Oder anders gesagt, es besteht der Bedarf für den Einsatz bandbreitenschonender Techniken.

3.3.9 Konfiguration des Systems

Die Konfiguration des Laufzeitsystems besteht hauptsächlich aus der Kalibrierung von Tracking- und Displaysystemen. Diese Aufgaben sollten durch zusätzliche Dienstprogramme effizient zu bewerkstelligen sein, da sie größtenteils nur einmalig für jedes System durchgeführt werden müssen. Lediglich die Anpassung von See-Through-HMDs muss bei jedem Aufsetzen des Displays neu angepasst werden. Dies sollte im Laufzeitsystem mit wenigen Schritten interaktiv oder besser halbautomatisch zu bewerkstelligen sein.

3.3.10 Erstellung von Inhalten

Die Hauptaufgabe bei der Vorbereitung einer AR-Anwendung ist typischerweise die Erstellung der Inhalte, die dem Anwender dann im Laufzeitsystem präsentiert werden. Der Erfolg einer AR Anwendung hängt ganz entscheidend von diesem Schritt und der Qualität der dadurch generierten Daten ab. Aber auch der Authoringaufwand ist dabei zu beachten, da dieser besonders bei selten durchgeführten Anwendungen leicht die Kosten-Nutzen-Bilanz des Einsatzes von AR Technologien zum Negativen verändern kann.

3.4 Nichtfunktionale Anforderungen

Tests und Befragungen bei potentiellen Endanwendern in Produktion und Service haben ergeben, dass vor allem nichtfunktionale Aspekte eines mobilen AR-Systems dessen Akzeptanz beeinflussen. Die überwiegende Zahl dieser Anforderungen betrifft die Entwicklung von besser geeigneter mobiler Hard-

ware, und wird deshalb nicht im Rahmen dieser Arbeit betrachtet. Aus Gründen der Vollständigkeit werden diese Anforderungen hier im Folgenden aufgezählt.

3.4.1 Qualität des Displays

Die Qualität und Bedienbarkeit von Displaysystemen beeinflusst die Akzeptanz und Performanz der Benutzer eines AR basierten Assistenzsystems [WIED01]. Bedeutet das Ablesen der Informationen im Display zusätzliche Belastungen für den Anwender, wird er auch eine gute Assistenzfunktionalität nur bedingt als Hilfe empfinden. Marktverfügbare HMDs weisen hier noch viele Schwächen wie hohes Gewicht, wenig Tragekomfort und nicht ausreichende optische Eigenschaften auf.

3.4.2 Performanz

Da die hauptsächlich betrachtete Anwendungsdomäne die Mobilität von Benutzer und AR System verlangt, muss auch auf entsprechend mobile Recheneinheiten zurückgegriffen werden. Auch, wenn mittlerweile Notebooks mit Leistungsmerkmalen von Desktopsystemen verfügbar sind, können diese aufgrund von hohem Gewicht, Bauform und hohem Stromverbrauch nicht als wirklich mobil betrachtet werden. Rechner mit wirklich mobilen Abmessungen und Gewicht, so genannte Wearables, können dagegen bei weitem nicht mit Prozessor- und Grafikleistung stationärer Rechner mithalten. Beides wichtige Kriterien, da die meist eingesetzten bildbasierten Tracking-Verfahren viel Rechenlast verursachen und auch das Rendering der virtuellen Szene 3D-Grafikunterstützung erfordert, um auch auf mobiler Hardware hohe Bildwiederholraten und kurze Latenzzeiten zu erreichen. Speziell Rechner der Wearable- oder PDA-Klassen liefern hier nur unbefriedigende Leistungsdaten. Um dieses Dilemma zu umgehen, wurden Systeme entwickelt, die Tracking und Rendering auf einem zusätzlichen stationären Rechner durchführen und Video- sowie Renderbild über ein drahtloses Netzwerk übertragen [PASM03]. Allerdings erfordert dieser Ansatz zusätzliche Hardware, eine stabile WLAN-Verbindung mit hoher Bandbreite und verursacht zusätzliche Latenz.



Abbildung 18: Typische Forschungsprototypen mobiler AR Systeme (v.l.n.r. Tinmith-evo5, Studierstube, DWARF)

3.4.3 Gewicht und Größe

Wie bereits im vorherigen Punkt angedeutet, handelt es sich bei großer Leistungsfähigkeit und geringem Gewicht sowie kleiner Bauform mobiler Rechner um offensichtlich konkurrierende Ziele. Mit heutigen marktverfügbaren Geräten lassen sich beide Ziele gleichzeitig nicht erfüllen, so dass der Hardwareaufbau aktueller Forschungsprototypen (siehe Abbildung 18) vor allem dafür herangezogen werden kann den Bedarf an entsprechender Hardware zu verdeutlichen.

Soll ein Servicetechniker oder ein Monteur mit AR Unterstützung versehen werden, muss dieser das System mehrere Stunden ununterbrochen mit sich führen. Ein großer und schwerer Rucksack erscheint hier nicht akzeptabel.

3.4.4 Betriebsdauer / Stromversorgung

Obwohl kleine, leistungsstarke Stromversorger für mobile Systeme verfügbar sind, summiert sich bei einem kompletten System bestehend aus Rechner, Display, Kamera, eventueller zusätzlicher Tracking-Sensorik und sonstigen Peripheriegeräten der Energieverbrauch nicht unbeträchtlich auf. Die verfügbare Einsatzzeit nimmt dadurch ab und muss durch zusätzliche Stromquellen ausgeglichen werden, die wiederum zusätzliches Gewicht verursachen.

3.4.5 Sonstige Anforderungen

Ein wichtiger Aspekt, der sich teilweise aus den vorherigen Punkten ableiten lässt, ist die Sicherheit bei der Verwendung solcher Systeme. Obwohl Augmented Reality per Definition den Kontakt des Benutzers zu seiner Umgebung bestehen lässt, sieht dies in der Realität meist anders aus. Sowohl optische als auch Video See-Through-Displays besitzen prinzipbedingte Schwächen, so dass die Auswahl eines Displaysystems auf eine spezielle Anwendung abgestimmt werden muss. Das optimale Displaysystem, das alle Disziplinen gleichermaßen gut beherrscht, existiert nicht.

3.5 Bestehende Systeme

In diesem Abschnitt werden bestehende AR-Systeme und ihre Systemarchitekturen betrachtet. Diese Betrachtung soll vor allem unter zwei Aspekten geschehen: der Systemarchitektur und der Berücksichtigung eines kompletten AR-Prozesses bestehend aus dem Erstellen der Inhalte, der Konfiguration des Systems und dem eigentlichen Laufzeit-System.

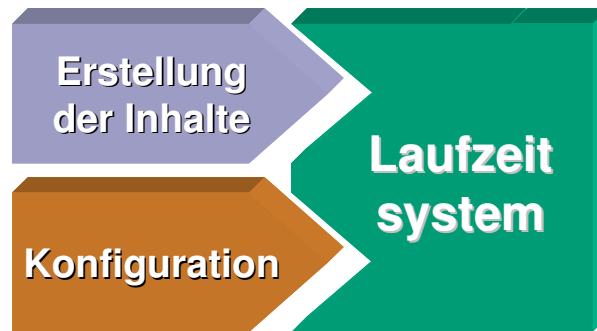


Abbildung 19: Prozesssicht

3.5.1 ARToolkit

Das wohl am weitesten verbreitete, frei verfügbare System ist das **ARToolKit**. Es handelt sich dabei um eine Softwarebibliothek für die schnelle Entwicklung von prototypischen MR- und AR- Anwendungen, sowie Tracking- und Interaktionstechniken. Verfügbar ist das ARToolKit auf den Plattformen Windows und Linux. Das Tracking wurde auch auf Windows basierte PDAs portiert. Speziell für Forschungsarbeiten, Anwendungen und Erweiterungen basierend auf dem ARToolKit wird seit 2002 ein eigener Workshop veranstaltet.

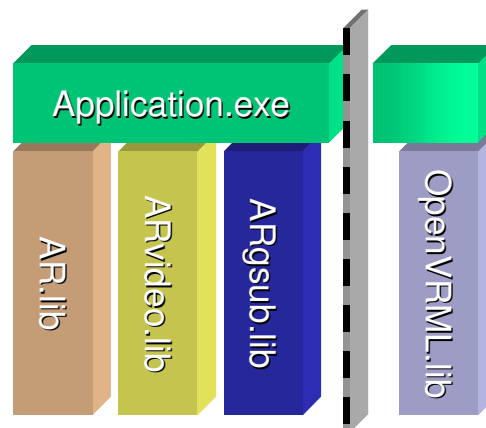


Abbildung 20: ARToolKit Architektur

Das Toolkit besteht aus drei Bibliotheken (siehe Abbildung 20), auf deren Funktionalitäten eine Applikation aufsetzt. Die Bibliotheken stellen die Funktionalitäten Tracking (AR.lib), OpenGL-Rendering (ARgsub.lib) und Zugriff auf Video-Quellen (ARvideo.lib) bereit. Ergänzend zum reinen OpenGL-Rendering kann auch die OpenVRML-Bibliothek eingebunden werden, so dass dem VRML97 Standard entsprechende Szenen geladen werden können.

Oft wird die Tracking-Funktionalität aus dem Gesamtsystem herausgelöst verwendet. Dies erfordert eine objektorientierte Abstraktion des in C geschriebenen Tracking Codes. Haller et al [HALL02] stellen basierend auf diesem Ansatz die direkte Integration des ARToolKit Trackers in die szenen-graphbasierten Renderer OpenSG und OpenSceneGraph vor. Darüber hinausgehend steht der ARToolKit Tracker auch als Tracking-Modul in der Geräteschnittstelle OpenTracker zur Verfügung und ist somit auch in der Studierstube Umgebung verfügbar. Die mit dem Studierstube Rahmensystem realisierten mobilen Augmented Reality Anwendungen verwenden das ARToolkit Tracking-Modul für

die Verfolgung von Interaktionsgeräten [REIT01a]. Wagner und Schmalstieg haben den ARToolKit Tracker auf einen PocketWindows basierten PDA portiert und für diese Plattform optimiert [WAGD03].

Das ARToolkit bietet sich als Basis für Interaktions- und Tracking-Experimente an, besitzt aber über das Tracking hinausgehend keine Funktionalitäten, die bei der Realisierung von „echten“ Anwendungen notwendig sind. Es kann somit weder als Plattform noch als ernstzunehmendes Rahmensystem betrachtet werden.

Das ARToolkit stellt lediglich die Laufzeitfunktionalitäten zur Erstellung prototypischer AR Applikationen bereit. Da die im System enthaltene Darstellungsbibliothek direkt auf OpenGL basiert, können hier auch keine Standardwerkzeuge zur Szenenerstellung verwendet werden. Dies ändert sich, wenn das System in Verbindung mit OpenVRML verwendet wird, wodurch zumindest die Erstellung einfacher Applikationen in Standard 3D Software möglich wird. Durch den Fokus auf das videobasierte Tracking bei der Entwicklung des ARToolkit, besitzt dieses einen Satz zusätzlicher Tools zur Erstellung von Markern und zum Kalibrieren von Kameras. Im Rahmen darauf aufbauender Arbeiten sind viele weitere Tools entstanden, die zumindest die Einbindung des Tracking-Systems in andere Systeme erleichtern, die wiederum den Fokus auf die Prozesssicht setzen.

3.5.2 Studierstube

Die Augmented- und Mixed-Reality Plattform Studierstube [SCHMAL00] wurde ursprünglich für kollaborative Mixed Reality Anwendungen in Laborumgebungen entworfen. Im Laufe der Zeit wurde das System aber dahingehend erweitert, auch als Basis für mobile Augmented Reality Anwendungen zu dienen.

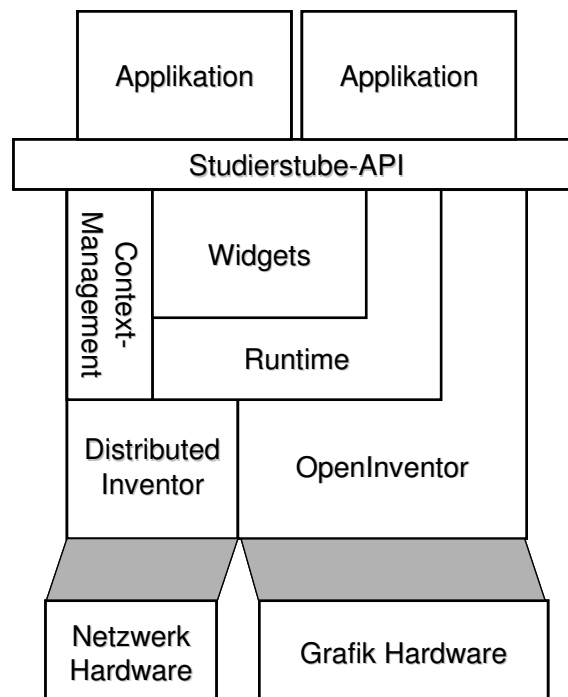


Abbildung 21: Studierstube Systemarchitektur [SCHMAL00]

Die Systemarchitektur basiert vollständig auf OpenInventor Szenengraph-Knoten [WERN93]. Neben den 3D Geometrien beinhaltet der Graph auch die komplette Applikationslogik sowie sämtliche Anwendungsdaten. Ziel dieser Architektur ist ein einheitliches Modell für die Datenhaltung, sowohl für 3D Geometrien als auch für alle anderen Daten.

Neue Funktionalitäten werden im Studierstube System durch die Erweiterung von OpenInventor, speziell durch die Implementierung neuer Knotentypen, realisiert. Eine verteilte Variante von OpenInventor, Distributed-Inventor, ermöglicht die einfache Verteilung der Anwendung auf mehrere Rechner. Der Szenengraph wird dabei auf den jeweiligen Rechnern repliziert. Da die Applikation selbst Teil des Szenengraphen ist, läuft somit auf jedem Host eine replizierte Instanz der Applikation.

3.5.3 DWARF

Das am Institut für Informatik der TU München entwickelte Framework DWARF (*Distributed Wearable Augmented Reality Framework*) [BAUER01], basiert auf der Verteilung von Diensten (Services) auf so genannte Module. Ein DWARF-Modul beinhaltet sowohl die Software, als auch die für einen Service notwendige Hardware. Diese Kombination von Soft- und Hardware realisiert die so genannte Tool-Metapher. Die Funktionalitäten des Systems werden somit für den Benutzer greifbar gemacht. Ein Tracking-Modul für videobasiertes Tracking beinhaltet neben den Tracking-Algorithmen auch die Recheneinheit und die Kamera. Möchte der Anwender videobasiertes Tracking verwenden, greift er die Kamera, verbindet sie mit dem DWARF-System und aktiviert sie.

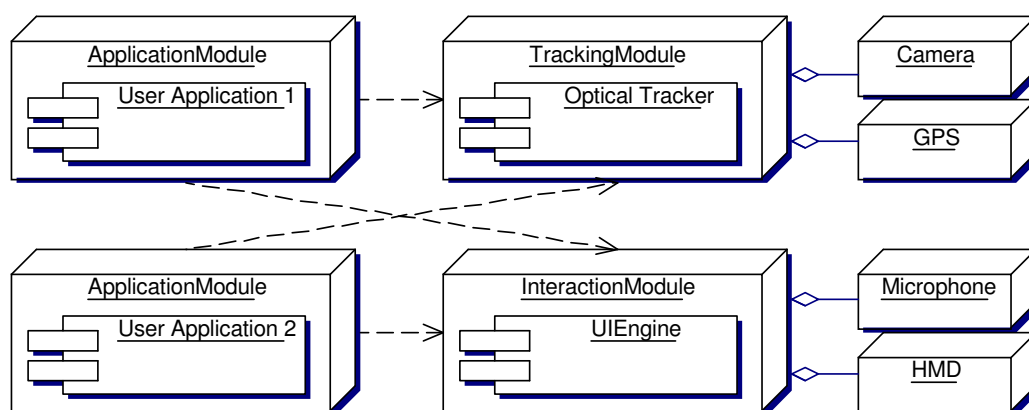


Abbildung 22: DWARF Module, bestehend aus Soft- und Hardware

Jeder Dienst, den ein Modul anbietet, wird durch seine Fähigkeiten (Abilities) und Bedürfnisse (Needs) beschrieben. DWARF Dienste können sich zur Laufzeit dynamisch verbinden, nachdem die Abilities und Needs verglichen wurden. Über eine CORBA Middleware werden die Dienste lose miteinander gekoppelt. Die Verwaltung des Gesamtsystems erfolgt dabei komplett dezentral über so genannte Servicemanager. Jedes Modul besitzt einen solchen Servicemanager, der die lokalen Services kontrolliert und ihre Beschreibung für andere Services bereitstellt. Wird ein neuer Service im System aktiviert, stellt der Servicemanager des betreffenden Moduls Verbindungen zu den anderen Services

her. Er versucht nun seine Bedürfnisse mit den angebotenen Fähigkeiten anderer Services zu befriedigen. War dies erfolgreich, kann er selbst seinen Service anbieten oder bleibt ansonsten inaktiv.

Kernkonzept von DWARF ist die transparente Verteilung der Dienste auf unterschiedliche Rechner, so dass auch Dienste in der Umgebung des Benutzers einfach genutzt werden können. Begibt sich der Benutzer in den Einflussbereich eines Tracking-Systems, dass in einem Raum installiert ist, kann sich der damit verbundene Tracking-Service beim DWARF-System des Benutzers anmelden und dort seine Dienste anbieten. Das DWARF-System stellt somit eine Brücke zwischen Augmented Reality und Ubiquitous Computing dar [BAUER02].

3.5.4 Tinmith-Evo5

Die Architektur von Tinmith-evo5 [PIEK01a] wird durch die Datenflüsse von den Tracking Sensoren zum Rendering bestimmt. Im Vordergrund beim Entwurf des Systems stand vor allem die Performanz von Tracking und 3D Rendering. Die Erfassung und Verarbeitung des Kontexts und die Bereitstellung von kontextbezogenen Informationen fanden dabei keine größere Beachtung.

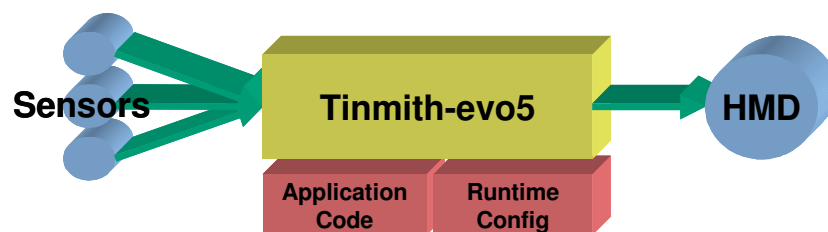


Abbildung 23: Tinmith-evo5 System-Architektur

So wird in Tinmith die Serialisierung von Objekten in XML-Dateien unterstützt. Beim Shutdown kann der Systemzustand festgehalten und beim Neustart an der gleichen Stelle aufgesetzt werden. Der Mechanismus der Serialisierung wird auch bei der Verteilung der Anwendung verwendet.

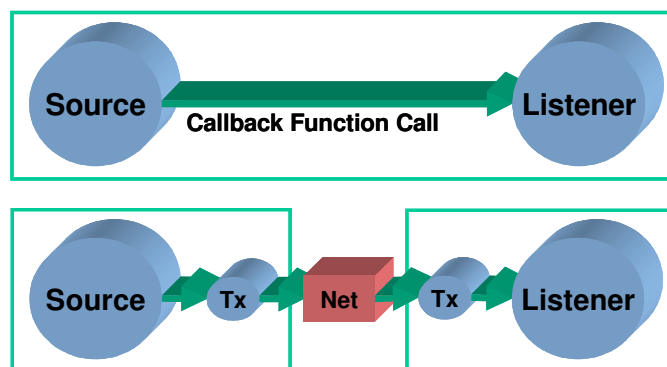


Abbildung 24: Serialisierungsobjekte sorgen für Netzwerktransparenz

Spezielle Serialisierungs- und De-Serialisierungs-Objekte machen den Netzwerkzugriff transparent (siehe Abbildung 24). Ganze Objekte des Systems können somit über ein Netzwerk versendet werden, ohne dass die Applikation Kenntnis davon haben muss, dass und wie sie auf mehrere Rechner verteilt ausgeführt wird.

3.6 Zusammenfassung

Bei der Realisierung eines Rahmensystems zu Erstellung von AR Anwendungen sind zuallererst die Anforderungen an ein solches System zu erfassen und geeignet aufzubereiten. Dabei werden vor allem die Anforderungen durch die verschiedenen Anwendungsgebiete betrachtet aber ebenso die verschiedenen Benutzergruppen berücksichtigt, die mit dem System in Kontakt kommen.

Basierend auf diesen Anforderungen erfolgt anschließend eine Analyse mit dem Ergebnis, einer Aufstellung und Beschreibung für mobile AR Anwendungen notwendiger Funktionalitäten. Neben diesen funktionalen Anforderungen werden auch nichtfunktionale Anforderungen, vor allem an die einzusetzende Hardware abgeleitet.

Mit dem Fokus auf die obigen Anforderungen werden bestehende AR Systeme und deren Architekturen betrachtet und bewertet, so dass in das im folgenden Kapitel vorgestellte Architekturkonzept für eine mobile AR Plattform weitestgehend alle Informationen, Anforderungen und Erfahrungen die in diesem Bereich verfügbar sind, einfließen können.

4 Architekturkonzept für eine mobile AR Plattform

Augmented Reality kann als Technologie betrachtet werden, die als Basis zur Realisierung unterschiedlichster Anwendungen verwendet wird. Ein AR System sollte demzufolge möglichst viele der aus diesen Anwendungen entstehenden Anforderungen erfüllen können oder zumindest einfach um zusätzlich benötigte Funktionalitäten erweiterbar sein. Idealerweise stellt ein Basissystem einen Satz von Funktionalitäten bereit, aus denen die Benötigten ausgewählt und angepasst werden können. Eine Anwendung besteht dann aus einem Satz von Basisfunktionalitäten, eventuellen Erweiterungen und den applikationsspezifischen Daten. Im folgenden Abschnitt wird eine System-Architektur vorgestellt, die eine nach dem obigen Prinzip gestaltete Erstellung von Anwendungen unterstützt.

4.1 Framework

Die Konzeption eines Rahmensystems, das der Realisierung verschiedenster Anwendungen als Basis dienen soll, kann als ein sehr komplexes Problem angesehen werden. Dies gilt im Besonderen, da im Falle von mobilem Augmented Reality viele der potentiellen Anwendungsfelder und deren spezifischen Anforderungen zum heutigen Zeitpunkt nur schwer absehbar sind.

Das Rahmensystem selbst kann bereits als Mittel zur Verringerung der Komplexität des Problems angesehen werden, da es ermöglicht, die Erfüllung von Anforderungen an eine spezifische Anwendung zu delegieren. Es müssen also keinesfalls alle Anforderungen bereits durch das Rahmensystem abgedeckt werden, sondern es soll vielmehr effizient bei der Erfüllung von Anforderungen unterstützen. Spezielle Funktionalitäten und Erweiterungen bestehender Funktionalitäten sollten schnell realisierbar sein.

Ein Schlüsselkonzept bei der Konzeption eines Rahmensystems stellt die Wiederverwendbarkeit von Software dar. Diese Wiederverwendbarkeit kann auf verschiedenen Ebenen realisiert werden, auf Code-Ebene, auf Komponenten-Ebene und auf der Basis ganzer Programme.

- Die Wiederverwendung auf Code-Ebene findet in der objektorientierten Entwicklung auf Basis von Klassen statt. Der Anwendungsentwickler kann sich aus einer Bibliothek von Klassen diejenigen aussuchen, die ihm bei der Lösung seiner Aufgabe helfen. Neue Klassen, die für ein spezielles Problem entworfen wurden, werden der Bibliothek hinzugefügt und stehen zukünftig bei der Realisierung von Anwendungen zur Verfügung. Eine Änderung an der Anwendung oder an deren Konfiguration erfordert allerdings ein neuerliches Übersetzen des Programmcodes.
- Wird die Wiederverwendung von Funktionalitäten auf der Basis von Komponenten realisiert, so kann die Konfiguration der Anwendung auch ohne ein neuerliches Übersetzen des Pro-

grammcodes geändert werden. Lediglich, wenn eine benötigte Funktionalität mit den bestehenden Komponenten nicht abgedeckt werden kann, muss eine neue Komponente erzeugt werden.

- Wiederverwendung auf der Ebene ganzer Programme bedeutet, dass eine Applikation unverändert für unterschiedliche Anwendungsgebiete eingesetzt werden kann. Allerdings ist typischerweise die Wiederverwendung auf eine relativ kleine Anwendungsdomäne mit ähnlichen Anforderungen begrenzt.

Natürlich können auch die unterschiedlichen Ebenen von Wiederverwendbarkeit in einer Anwendung komplementär genutzt werden. Dies geschieht beispielsweise in einer komponentenbasierten Applikation, in deren Komponenten vor allem Utilityklassen wiederholt verwendet werden.

Dem in dieser Arbeit beschriebenen Rahmensystem liegt ein Komponentenkonzept zu Grunde, dass ebenso die schnelle Evaluierung neuer Basistechnologien im Gesamtsystem erlaubt, wie es den Aufbau komplexer, spezialisierter Anwendungen unterstützt. Den Anforderungen angepasste Anwendungen können mit wenig Aufwand aus einem Pool bestehender Komponenten zusammengestellt und um eventuelle anwendungsspezifische Funktionalitäten ergänzt werden.

Voraussetzung für die Erstellung eines solchen komponentenbasierten Systems ist die Zerlegung der Gesamtaufgabe in kleine Teilaufgaben, die den Komponenten zugewiesen werden können. Eine Komponente stellt somit einen Dienst zur Verfügung, der von anderen Komponenten genutzt werden kann.

Angesprochen wird die Komponente dabei über eine genau definierte Schnittstelle. So ist es möglich, dass aus mehreren Realisierungen einer Funktionalität gewählt werden kann, indem Komponenten mit gleicher Schnittstelle gegeneinander austauschbar sind. Grundvoraussetzung für die Definition von Schnittstellen, die für eine ganze Gruppe von Funktionalitäten Gültigkeit besitzen, ist die Abstraktion der damit angesprochenen Funktionalitäten.

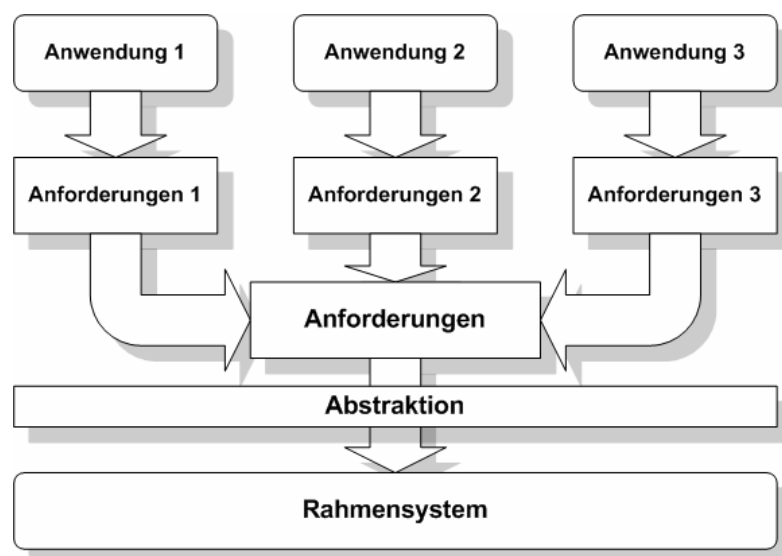


Abbildung 25: Vorgehensmodell zur Erstellung eines Rahmensystems

Ein Beispiel für eine solche Abstraktion ist eine Tracking-Komponente, die unabhängig von ihrer spezifischen Realisierung der Applikation immer den gleichen Dienst anbietet. Für die Anwendung bleibt

es so transparent, welches Tracking-Verfahren eingesetzt wird. Eine Anwendung kann über den Austausch der Tracking-Komponente an unterschiedliche Umgebungen angepasst werden, ohne dass die Anwendung selbst verändert werden muss.

Eine der grundlegendsten Anforderung an ein Augmented Reality System ist dessen maximale Performanz auf einer gegebenen Hardwareplattform. Hier ist besonders die Realisierung von Basisfunktionalitäten, wie Rendering und Tracking betroffen, die in besonderem Maße Rechnerressourcen wie Prozessor und Grafik belasten.

Neben der Betrachtung einzelner Komponenten darf aber auch deren Zusammenspiel nicht vernachlässigt werden. Hier ist zu beachten, dass Performanz und Flexibilität des Systems konkurrierende Ziele sind. Größere Flexibilität wird über ein höheres Maß an Kommunikationsaufwand erkaufte. Eine Middleware wie CORBA oder COM erlaubt zwar zusätzlich das transparente Verteilen der Komponenten auf verschiedene Rechner, diese Möglichkeit kann allerdings in den meisten realen Anwendungsszenarien nur sehr bedingt genutzt werden, so dass dies einen eher theoretischen Vorteil bedeutet.

Anders als in ereignisgesteuerten Desktop-Anwendungen wird bei einem AR System eine Hauptschleife kontinuierlich durchlaufen. Jedes Durchlaufen dieser Hauptschleife beschreibt einen Frame also die Präsentation der Informationen registriert mit der realen Umgebung, basierend auf den aktuellsten verfügbaren Tracking-Daten. Jede Verzögerung an dieser Stelle wird als Fehler in der Registrierung oder als Verzögerung der virtuellen gegenüber der realen Welt wahrgenommen und beeinträchtigt somit die Akzeptanz durch den Anwender.

In jedem einzelnen Frame wird die aktuellste Position und Blickrichtung im Tracking-System berechnet und anschließend an die Präsentationskomponente weitergegeben. Zusätzlich fließen hier noch durch etwaige Benutzerinteraktionen ausgelöste Änderungen der Szene ein. Sind alle Änderungen verarbeitet, kann die virtuelle Szene dargestellt werden.

Während also Benutzerinteraktionen nur sporadisch zu verarbeiten sind, sind Tracking und Präsentation in jedem Frame durchzuführen. Während selbst Standard-Desktoprechner mittlerweile mit leistungsfähiger Grafikbeschleunigung ausgestattet sind, besitzen Rechner unterhalb der Notebook-Klasse wenn überhaupt meist nur über rudimentäre 3D-Grafikunterstützung. Entsprechend viel zusätzliche Prozessorlast wird auf solchen Rechnern durch die Darstellung der virtuellen Szene verursacht. Der größte Anteil der verbrauchten Rechenzeit pro Frame entfällt allerdings bei der Verwendung visionbasierter Tracking-Verfahren auf die damit verbundene Bildverarbeitung. Die für Funktionalitäten des Frameworks zur Verfügung stehende Rechenzeit ist also dementsprechend gering und muss effizient genutzt werden.

Noch entscheidender als die durch hohe Prozessorlast verursachten niedrigen Frameraten sind in diesem Zusammenhang die Verzögerungen zwischen dem Beginn eines Frames bis zur endgültigen Präsentation. Diese äußern sich in Optical-See-through HMDs als dynamische Registrierungsfehler, als Nachziehen der virtuellen Szene gegenüber der realen Umgebung oder in Video-See-Through Displays als Verzögerung der kombinierten realen und virtuellen Welt. Die Datenflüsse innerhalb des Systems sind so optimal zu gestalten, dass hier zumindest keine zusätzlichen Delays entstehen.

Trotz der primären Forderung nach Performanz ist bei der Konzeptionierung des Frameworks auch dessen Flexibilität von besonderer Bedeutung. Dies beinhaltet die einfache Austauschbarkeit von Komponenten ebenso wie weit reichende Möglichkeiten der Konfiguration des Frameworks und der darin enthaltenen Komponenten.

4.2 System-Architektur

Ziel der hier vorgestellten System-Architektur ist, das sie die einfache Realisierung von Anwendungen ermöglicht, die die in Abschnitt 3 gesammelten Anforderungen an solche Applikationen erfüllen. Die aus den Anforderungen abgeleiteten Funktionalitäten gilt es dabei als Funktionalität des Basissystems zu realisieren oder, wo dies angebracht ist, die Realisierung der Funktionalität in der Anwendung durch Bereitstellung von entsprechenden Basisfunktionalitäten zu unterstützen.

4.2.1 Übersicht

Die Architektur des AR Basissystems kann in drei Schichten unterteilt werden, eine systemnahe Schicht, in der alle plattformspezifischen Funktionalitäten angesiedelt sind, die Kernfunktionalitäten des Frameworks in der mittleren Schicht und schließlich die Applikationsschnittstellen, die eine Integration der Kernfunktionalitäten in eine Anwendung erlauben (siehe Abbildung 26).

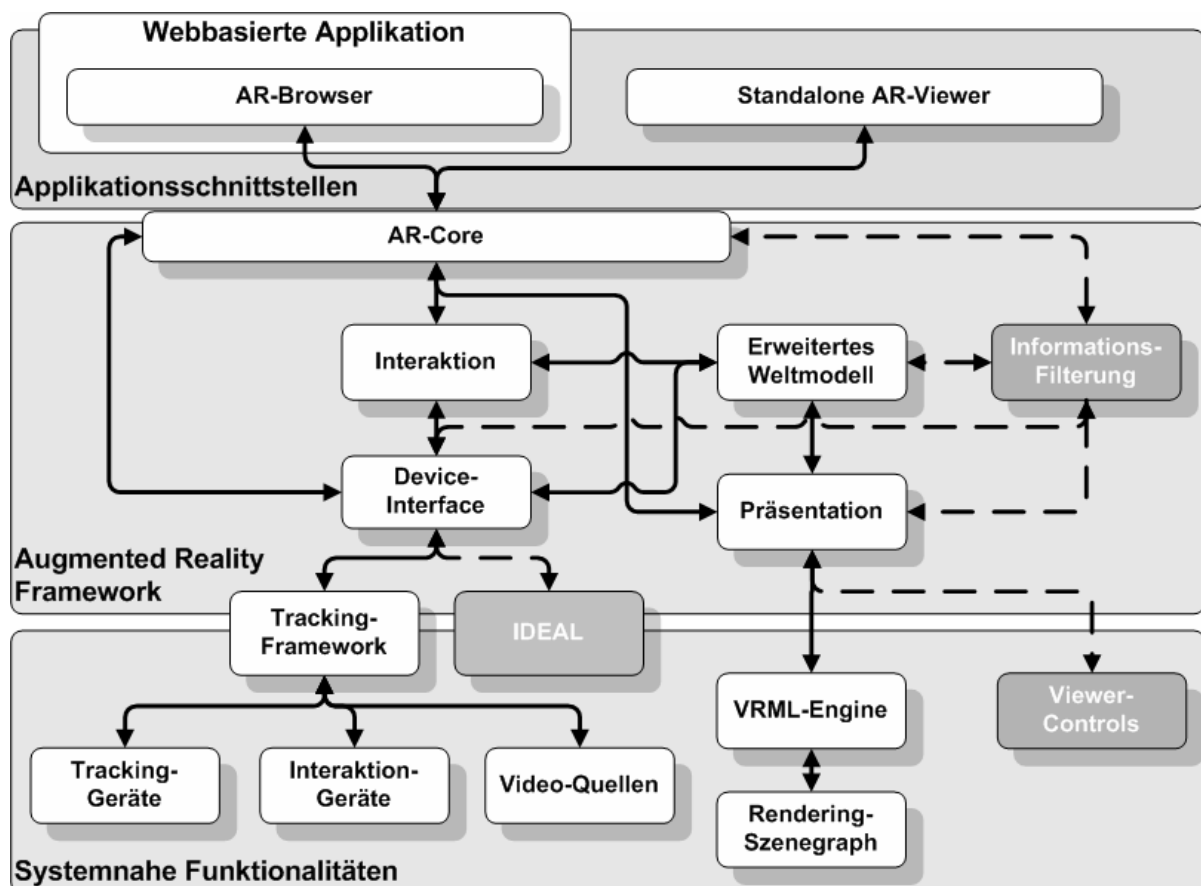


Abbildung 26: Rahmensystem für die Realisierung mobiler AR Anwendungen

Die systemnahen Funktionalitäten, wie das Bereitstellen von Video-Bildern oder der Zugriff auf Tracking- und Interaktionsgeräte sind in der untersten Schicht gekapselt. Die spezifischen Ausprägungen dieser Funktionalitäten können somit den Gegebenheiten unterschiedlicher Plattformen angepasst werden. Auch die eigentliche Grafikausgabe mittels VRML-Engine und unterliegendem Szenengraphen sind auf dieser Schicht angesiedelt. Obwohl diese Funktionalitäten selbst auf Portabilität ausgerichtet sind, werden jedoch Plattformen wie PDAs oder Smartphones, die gerade für den mobilen Einsatz von AR interessant sind, nicht direkt unterstützt, da hier auf den unteren Schichten, wie OpenGL bzw. DirectX große Unterschiede zu vollwertigen Rechnern bestehen. Auch die Viewer-Controls befinden sich auf der systemnahen Schicht. Über diese Komponenten können Viewer für Standard-Dokumenttypen, wie Pdf oder Quicktime eingebunden und angesprochen werden.

In der mittleren Schicht befinden sich die eigentlichen Kernfunktionalitäten, wie Interaktion und die Präsentations-Komponente. Der Zugriff auf Geräte wird auf dieser Schicht über eine einheitliche Geräteschnittstelle durchgeführt. Neben dem Tracking-Framework, das der Komposition kombinierter Tracking-Verfahren dient, können über diese Schnittstelle auch andere Geräteschnittstellen, wie Ideal oder OpenTracker angebunden werden.

Das erweiterte Weltmodell erlaubt losgelöst von der geometrischen Repräsentation eines realen Objekts im Szenengraphen die Anreicherung der realen Umgebung des Benutzers mit virtuellen Informationen. Diese Informationen und auch damit assoziierte Szenengraphknoten können mit zusätzlicher Semantik versehen werden. Das Weltmodell stellt somit eine dreidimensionale Schnittstelle für den Zugriff auf Informationen, die mit realen Objekten verbunden sind, zur Verfügung. Eine der Komponenten, die mit der zusätzlichen Semantik in den Informationsobjekten des Weltmodells arbeitet, ist die Informationsfilterung, die auf Basis dieser Informationen die Darstellung der Objekte steuert.

Angesprochen werden können die Komponenten dieser Systemschicht über eine einheitliche Programmierschnittstelle. Auf diese Schnittstelle aufsetzend erlaubt das AR-Browser Control die Integration von Augmented Reality Inhalten in Webbasierte Anwendungen.

4.2.2 Framework Basisfunktionalität

Die Komponenten des Frameworks lassen sich grundsätzlich in zwei Gruppen einteilen, in Komponenten die Basisfunktionalitäten realisieren und solche, die zusätzliche Dienste anbieten. Da das Framework eine abgeschlossene Applikation darstellt, können diese Komponenten als C++ Klassen realisiert werden. Auf eine aufwändige Middleware kann so mit Blick auf die Verwendung eines mobilen Rechners und dessen beschränkte System-Ressourcen verzichtet werden. Komponenten die im Framework einen Dienst anbieten, erlauben den Zugriff auf die damit verbundene Funktionalität über ein definiertes Interface, dass allen anderen Komponenten bekannt ist.

Eine transparente Verteilung von Komponenten auf andere Rechner, wie im DWARF Framework [BAUER01], wird so vom Framework nicht unterstützt. Die Verteilung einzelner Funktionalitäten liegt somit in der Verantwortlichkeit der Komponenten, die diese Funktionalitäten bereitstellen.

4.2.3 Komponenten

Die unterschiedlichen Funktionalitäten, die innerhalb des Frameworks angeboten werden, sind in den Komponenten gekapselt. Dabei hat jede Komponente einen eng umrissenen Aufgabenbereich, für den sie zuständig ist. Die Integration in das Framework und die Kommunikation zwischen den Komponenten wird durch Funktionalitäten und Eigenschaften unterstützt, die jede darin enthaltene Komponente besitzt. Diese zugesicherten Funktionalitäten und Eigenschaften lassen sich in drei Kategorien einteilen: Erstens eine genau definierte Schnittstelle zum Erzeugen, Initialisieren, Administrieren und Zerstören der Komponenten, zweitens eine Menge von Attributen, über die eine Komponente ihren Zustand nach außen präsentieren kann und als drittes die Unterstützung des frameworkweiten Eventmechanismus.

4.2.3.1 Methoden

Die Methoden der Schnittstelle dienen vor allem der Änderung des Komponentenstatus von einer Phase in die nächste des Lebenszyklus einer Komponente (siehe Abbildung 27).

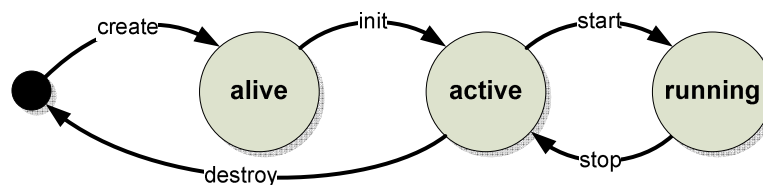


Abbildung 27: Zustände einer Komponente

Eine Komponente befindet sich danach immer in genau einem Zustand und kann von dort aus auch nur die gekennzeichneten Zustandsübergänge erreichen. Im Zustand „alive“ ist eine Komponente erzeugt, jedoch noch nicht initialisiert. Ein definierter initialer Zustand kann erst im Status „active“ zugesichert werden. Ist eine Komponente aktiv, läuft sie also in der Hauptschleife des Frameworks mit oder besitzt einen eigenen Ausführungsstrang, so kann sie weiter in den Zustand „running“ wechseln. Andere Komponenten, die lediglich als passive Server auf Anfragen von außen reagieren, verbleiben im Zustand „active“.

Neben diesen grundlegenden Methoden, die der Steuerung des Zustands einer Komponente dienen, besitzen die Komponenten, die einen Basisdienst zur Verfügung stellen ebenfalls eine an den jeweiligen Dienst angepasste, genau definierte Schnittstelle, die sie implementieren müssen. Zu diesen Basisdiensten gehört das Bereitstellen von Videobildern oder das Bereitstellen von Transformationen.

4.2.3.2 Attribute

Da sich die Komponenten grundsätzlich untereinander nicht kennen und auch außer den Basisdiensten nichts von der Aufgabe einer anderen Komponente wissen, können Komponenten ihren Zustand über eine Menge von Attributen nach außen kommunizieren. Ein Attribut besteht dabei jeweils aus seinem für diese Komponente eindeutigen Bezeichner und dem eigentlichen Wert. Ein Attribut besitzt dabei einen Typen, der dem Typ des Wertes entspricht und kann auch nur Werte dieses Typen annehmen.

Der Satz an Attributen einer Komponente wird beim Übergang in den Zustand „alive“ angelegt und initialisiert. Je nachdem, ob es sich um ein Read- oder ReadWrite-Attribut handelt, kann von außen nur lesend oder auch schreibend darauf zugegriffen werden. Innerhalb des Frameworks kann auf ein Attribut über vollständige Qualifizierung zugegriffen werden, wobei sich der Pfad zu einem Attribut aus dem Bezeichner der Komponente und dem Bezeichner des Attributs zusammensetzt, also zum Beispiel „Videosever.imageWidth“.

Eine Komponente, die sich für den Zustand einer anderen Komponente interessiert, kann also versuchen sich mit den passenden Attributen zu verbinden und deren Werte auszulesen. Über ReadWrite-Attribute ist es zusätzlich möglich, den Zustand einer anderen Komponente von außen zu beeinflussen. Damit eine Komponente auf die Änderung eines Attributs reagieren kann, lässt sich für die Verarbeitung eines geänderten Wertes eine Callback-Methode registrieren, die sofort nach der Änderung aufgerufen wird.

Ein Problem im Vergleich mit der methodenbasierten Kommunikation zweier Komponenten ist, dass zu einem Zeitpunkt immer nur der Wert genau eines Attributs einer Komponente ausgelesen oder geschrieben werden kann. So kann sich der Zustand beim nächsten Schreiben oder Lesen eines anderen Attributs bereits geändert haben. Die beiden Werte ergeben also kein zusammenhängendes Bild eines einzelnen Zustands, sondern sind vielmehr Teil zweier unterschiedlicher Zustände. In diesem Fall wird also ein Übertragungsmodus für mehrere zusammenhängende Attribute benötigt, der eine Übertragung im Sinne einer Transaktion ermöglicht. Die komplette Transaktion soll dabei nur dann erfolgen, wenn auch alle einzelnen Attribute zur gleichen Zeit ausgelesen oder geschrieben werden können. Da die Komponenten aber unter Umständen in verschiedenen Threads laufen, muss eine Synchronisierung erfolgen, damit auch bei einer Transaktion von mehreren verschiedenen Komponenten ein einheitlicher Zustand erhalten wird. Hierfür werden die entsprechenden Attribute gelockt und erst, wenn alle beteiligten Attribute auf diese Weise gegen weitere Zugriffe geschützt werden konnten, wird die Transaktion ausgelöst. War ein solcher Versuch nach einer gewissen Zeitspanne erfolglos, werden die bisher erhaltenen Locks wieder freigegeben. So kann verhindert werden, dass es zu Deadlocks kommen kann, allerdings kann so auch nicht zugesichert werden, dass eine Transaktion bei ungünstiger Parametrierung jemals zu Stande kommt.

Da also auch trotz des Transaktionsmodus eine gewisse Unsicherheit bezüglich der Übertragung zusammenhängender Informationen besteht, werden verschiedene komplexere Datentypen, wie Transformationsmatrizen oder auch Videobilddaten zusammen mit Header in einem einzelnen Attribut bereitgestellt.

Neben der Vereinheitlichung der Kommunikation innerhalb des Frameworks zwischen den Komponenten vereinfachen die Attribute auch die Verteilung der Komponenten auf andere Rechner. Hierfür wird auf dem Rechner, der das Framework beheimatet, lediglich ein Stub für die Kommunikation mit der entfernten Komponente eingerichtet. Die Kommunikation zwischen Stub und Komponente erfolgt über das Versenden der serialisierten Attribute. Sowohl für die eigentliche Komponente als auch für das Framework bleibt die Verteilung dabei transparent.

4.2.3.3 Events

Während eine Komponente auf die Änderung ihrer Attribute über Callbackmethoden reagieren kann, ist es mit den bisher beschriebenen Mechanismen für eine andere Komponente nur über regelmäßiges Auslesen möglich, Zustandsänderungen eines bestimmten Attributs zu erkennen. Um dieses Problem zu beheben, können sich Komponenten als Interessenten für die Änderungen eines Attributs anmelden. Sobald sich der Zustand des Attributs ändert, wird an alle Komponenten, die sich registriert haben, eine Nachricht gesendet, die die Änderung signalisiert. Die benachrichtigten Komponenten können nun entsprechend darauf reagieren.

4.2.4 Basisdienste

Generell ist die gewählte Architektur weniger generisch, als andere komponentenbasierte Architekturen. Allerdings sind bei der Realisierung von AR Systemen von vorne herein feste Strukturen und Beziehungen vorhanden, die einer zusätzlichen Generalisierung der Architektur entgegenstehen. Ein Tracking-System stellt als Leistung die aktuelle Position und Orientierung des Benutzers sowie weiterer getrackter Objekte zur Verfügung, unabhängig davon, wie das Tracking konkret realisiert ist. Jede andere Komponente kennt diesen Service und kann somit auf die Tracking-Komponente zugreifen, um den aktuellen Status abzufragen. Zusätzlich kann in einem AR System jede andere Komponente fest davon ausgehen, dass eine Tracking-Komponente vorhanden ist. Das gleiche gilt auch für die anderen Basisdienste, Präsentation und Interaktion, die Azuma in seiner Charakterisierung [AZUMA97] nennt.

Ein Beispiel dafür, wie sich zusätzliche Komponenten in das System einklinken, ist die Informationsfilterung, die an verschiedenen Stellen in das Basissystem eingreift. Die Informationsfilterung benötigt die Tracking-Informationen zur Bestimmung des aktuellen Standorts des Benutzers und seiner Blickrichtung. Zusätzlich besitzt die Informationsfilterung Zugriff auf das Weltmodell und greift auch auf die Attribute von AR-Core und die Methoden der Präsentation ein.

4.2.5 Erweitertes Weltmodell

Bei der Verwendung visionbasierter Tracking-Verfahren ist ein Modell notwendig, dass die reale Umgebung in einer für das jeweilige Verfahren adäquaten Form abbildet. Klinker prägt für diese Modelle den Begriff *Reality-Modell* [KLI98]. Diese Modelle enthalten die Informationen, die notwendig sind, um reale und virtuelle Welt zu registrieren. Hierbei kann es sich je nach verwendetem Tracking-Verfahren um die Beschreibung künstlicher Marker, speziell aufbereiteter Geometriemodelle oder auch Referenzbilder handeln. Während dieses dreidimensionale Modell der Szene speziell für die Verarbeitung im Tracking-System ausgerichtet ist, bilden auch die virtuellen Informationen, die dem Benutzer präsentiert werden, ein räumliches Modell. Hoellerer beschreibt in diesem Zusammenhang ein System zur Hypertext-Verlinkung von realen Orten und Objekten mit Multimedia-Objekten [HOELL99b].

4.2.5.1 Informationsraum

Die Menge der virtuellen Informationen die zu einem Zeitpunkt t für den Benutzer in seiner Umgebung verfügbar sind, wird als Informationsraum bezeichnet. Unabhängig davon, wie die Daten organisiert sind, sind sie in einer AR-Umgebung mit einem Ort oder einem Objekt der realen Umgebung verknüpft. Einzelne Informationen können somit direkt verortet werden. Die Ein- und Ausbauanleitung für ein Bauteil kann somit direkt über dessen realen, physikalischen Ort verfügbar gemacht werden. Der Einstieg in die Dokumentation einer komplexen technischen Anlage kann so sehr einfach über die somit entstandene räumliche Benutzerschnittstelle erfolgen, während der Anwender vor der Anlage steht. Ohne die Kenntnis passender Suchbegriffe hat ein Anwender sofort Zugriff auf die Informationen zu einem Objekt seiner realen Umgebung zur Verfügung.

Dem gegenüber stehen die aktuell gültigen Paradigmen der Datenverarbeitung und -verwaltung. Typischerweise findet man Informationen über entsprechende Suchbegriffe oder kann über die Verbindungen vernetzter Informationen zu den gewünschten Informationen navigieren. Der direkte räumliche Bezug zur realen Umgebung ist dabei nicht vorgesehen. Für die Verknüpfung bestehender Datenbestände in ihrer Organisationsform mit den intuitiven Zugriffsmöglichkeiten über ein AR Interface sind effiziente Strukturen und Vorgehensweisen zu schaffen. Dabei sollte der Eingriff in die bestehenden Strukturen so gering wie möglich gehalten werden, zumindest so lange die Informationen in einer Form vorliegen, die eine sinnvolle Darstellung auf mobilen Endgeräten erlaubt.

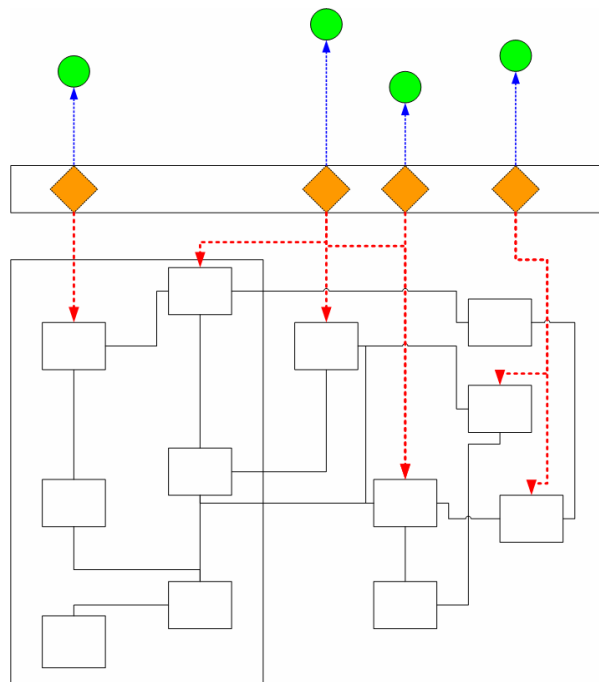


Abbildung 28: Verbindung von realen Orten und virtuellen Informationen

Um zu den gewünschten Informationen zu gelangen sind hierbei allerdings Interaktionen mit einer Benutzerschnittstelle notwendig, die eine Eingabe entsprechender Informationen erlaubt. Im Falle eines Fehlers einer modernen technischen Anlage ist dies noch relativ einfach zu bewerkstelligen, da hier meist ein Fehlercode ausgelesen werden kann, der auf das fehlerhafte Bauteil hinweist und für die automatisierte Bereitstellung relevanter Informationen verwendet werden kann. Tritt jedoch ein Fehler

auf, der nicht kategorisiert werden kann, greift auch dieses System nicht und der Anwender muss sich auf die Suche nach passenden Informationen begeben.

Noch deutlicher wird der Nutzen von AR Informationsräumen in Anwendungen, die im freien Modus durchgeführt werden, beispielsweise einem Informationssystem für eine archäologische Stätte. Hier kann sofort Vermittelt werden, für welchen Ort, für welches Objekt zusätzliche Informationen hinterlegt wurden. Der Besucher kann dann steuern für welches Objekt, das ihn interessiert, er zusätzliche Informationen möchte. Hierfür bieten sich auch andere Technologien an, die anhand des Ortes, an dem sich der Anwender befindet entsprechende Informationen zusammenstellen und präsentieren. Allerdings ist nur in AR der direkte räumliche Zusammenhang zwischen einem Objekt und den dazugehörigen Informationen so intuitiv für den Benutzer zu erkennen. Nur hier bilden reale Umgebung und virtuelle Informationen eine als solche wahrnehmbare Einheit.

Der Einstieg in die virtuellen Informationen erfolgt analog zur Auswahl einer Datei oder eines Links auf einer Desktop-Oberfläche über die Selektion des entsprechenden Objekts. Während dabei allerdings der Umweg über die Metapher der Desktop-Oberfläche und der darauf angeordneten Ordner und Objekte erfolgt, kann in der AR Umgebung die Selektion direkt über das verbundene reale Objekt erfolgen, was wiederum eine maximale Reiz-Reaktions-Korrespondenz ergibt, sobald der Anwender als Ausgabegerät ein See-Through-HMD trägt. Bei den meisten Video-Feed-Through-HMDs verringert sich die Reiz-Reaktions-Korrespondenz bereits, da hier eine Skalierung zwischen realer Umgebung und deren Abbild im HMD vorliegt, die bei der Interaktion vom Anwender berücksichtigt werden muss.

4.2.5.2 Informationsobjekt

Die kleinste Einheit bei der Modellierung von Informationsräumen stellen einzelne Informationsobjekte dar. Das Konzept der Informationsobjekte basiert auf der Beobachtung, dass zu einem Ort oder Objekt der realen Welt meist mehrere Informationen vorliegen. Begreift man AR als räumliches Interface zu diesen Informationen, sollten alle Informationen, die zu einem Objekt vorliegen, auch über dessen physikalischen Ort zugreifbar sein (Abbildung 29 a). Neben dieser 1 zu N Relation zwischen Ort und damit verknüpften Informationen gibt es auch den umgekehrten Fall, dass ein Informationsobjekt mit mehreren Orten verknüpft ist (Abbildung 29 b). Man denke hier zum Beispiel an gleichartige Bauteile, wie zum Beispiel Schrauben oder auch Räder, die mehrfach in einem Auto verbaut werden und sich nur in ihrem Einbauort unterscheiden. Verlässt man diese eher lokale Sichtweise und betrachtet gesamte Informationsräume so ergeben sich relativ eng vermaschte Netze aus Verbindungen zwischen Orten und dazugehörigen Informationen (Abbildung 29 c). Um das obige Beispiel weiterzuführen, liegen zu einem Rad des Autos mehrere Informationen vor.

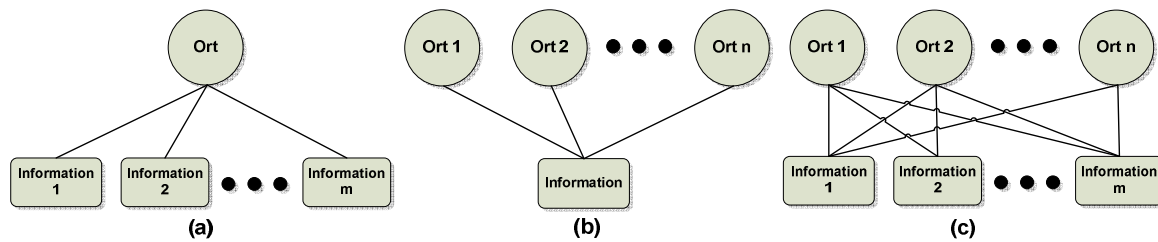


Abbildung 29: Relationen zwischen realen Objekten und Informationen

In rein virtuellen Umgebungen kann für die Verknüpfung von 3D Szene und Informationen das Geometriemodell der Szene selbst herangezogen werden. Das Geometriemodell kann hier soweit als Interface für den Zugriff auf Informationen verwendet werden, wie es die reale Szene ersetzt. Teile die nicht modelliert sind, können auch nicht sinnvoll Teil des Interfaces sein. Hier würden Informationen ohne sichtbare Beziehung im Raum hängen. In AR dagegen ist es nicht notwendig, die komplette Szene geometrisch zu rekonstruieren. Informationen können hier in 3D registriert werden und erhalten durch die reale Umgebung ihren korrekten räumlichen Kontext. Es kann somit unterschieden werden zwischen der Verbindung zwischen Informationen und Teilen der 3D Geometrie und der Verbindung zwischen Informationen und einem Ort für den keine Rekonstruktion der realen Szene vorliegt. In letzterem Fall ist es deshalb notwendig, eine einfache Platzhaltergeometrie an diesem Ort einzufügen, die später der Selektion in der virtuellen Szene dient.

Ist ein Geometriemodell der Szene vorhanden, ist dies meist nur bedingt für das direkte Verknüpfen von Informationen mit Geometrieteilen, die reale Objekte repräsentieren, geeignet. Hierbei gilt es zuerst die Herkunft und auch die ursprüngliche Aufgabe dieser Geometrien zu betrachten. Meist werden Geometriemodelle für die Visualisierung erstellt und sind dementsprechend auch für die möglichst performante Darstellung im 3D Renderer optimiert, was bedeuten kann, dass viele kleine Geometrien zu größeren Knoten zusammengefasst wurden, wodurch eine Identifizierung der Einzelobjekte im Modell nicht mehr möglich ist. Auch wenn die ursprüngliche Hierarchie des Modells noch vorhanden ist, bedeutet dies nicht, dass diese Struktur auch der Granularität der damit zu verknüpfenden Informationen entspricht. Vielfach befinden sich in 3D Modellen zusammengehörige Geometrien auf der gleichen Hierarchieebene, besitzen aber keinen zusammenfassenden Gruppenknoten. Da sich auf dieser Hierarchieebene auch beliebige andere Teilgeometrien befinden können, ist eine direkte Zuordnung zwischen Geometrien und realen Objekten quasi unmöglich (Abbildung 30, links). Eine Autotür, das Türfenster und der Türgriff liegen auf der gleichen Hierarchiestufe wie ein Kotflügel und die Motorhaube. An dieser Stelle wird eine Umstrukturierung der Szenengeometrie notwendig. Durch das Einfügen zusätzlicher Gruppenknoten kann die eindeutige Zuordenbarkeit zwischen Teilbäumen des Szenengraphen und realen Objekten hergestellt werden (Abbildung 30, links). Alle Teile der Autotür hängen jetzt unter einem gemeinsamen Gruppenknoten und sind somit als ein Objekt erkennbar. Zusätzlich sind die vorherigen Teilhierarchien erhalten geblieben, so dass auch das Schloss weiterhin einzeln mit Informationen verknüpft werden kann.

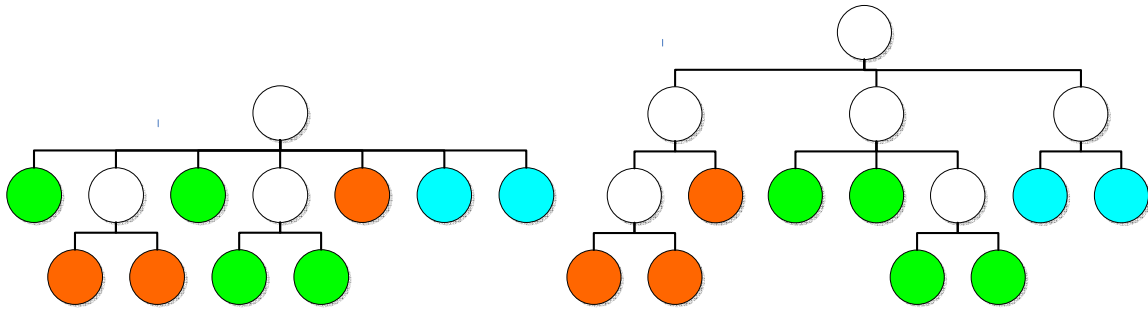


Abbildung 30: 3D Szenegeometrie, links vor, rechts nach der Umstrukturierung (Weiße Knoten repräsentieren Gruppenknoten, farbige Knoten signalisieren die Zugehörigkeit zu unterschiedlichen Objekten)

Liegt zu einem realen Objekt kein Geometriemodell vor, kann dies nachträglich durch 3D Scanner oder photogrammetrische Rekonstruktion erstellt werden. Allerdings fehlen hierbei Informationen über Hierarchie und Semantik vollständig und müssen nachträglich manuell erzeugt werden. In diesen Fällen, in denen die vorhandene 3D Szenenhierarchie nicht genutzt werden kann, um auf Objektebene Informationen damit zu assoziieren, ist das händische, sehr aufwändige Erzeugen von Geometrie-hierarchien notwendig.

Neben den Informationen an sich, die mit den Objekten verknüpft werden, sind zur gezielten Steuerung der Sichtbarkeit und der Aufmerksamkeit des Benutzers weitere Attribute zu Informationsobjekten notwendig (siehe Abbildung 31). Zur Implementierung der Attribute können hier die gleichen Mechanismen, wie bei den Attributen der Framework-Komponenten wieder verwendet werden. Eine Anwendung für solche zusätzlichen Attribute ist die Informationsfilterung, die bezüglich eines Zustandsvektors des Anwenders und eines korrespondierenden Attributvektors über die Darstellung eines Informationsobjekts entscheidet.

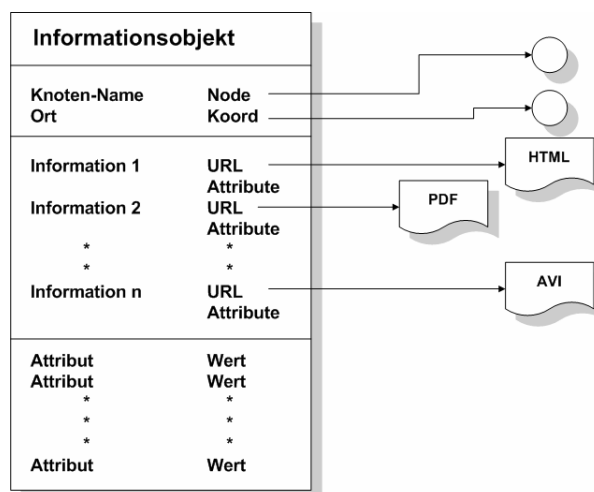


Abbildung 31: Struktur eines Informationsobjekts

Das Konzept der Informationsräume unter Verwendung von Informationsobjekten fügt der Anwendung eine weitere zusätzliche Datenstruktur neben dem Rendering-Szenengraph hinzu, die mit den anderen Daten synchronisiert werden muss. Diese Synchronisierung erfolgt über die Informationsobjekte selbst, die auch den einzigen Zugriffsweg auf die eigentlichen Präsentationsobjekte, wie 3D Ge-

ometrien darstellen. Eine Änderung einer Geometrie im Szenengraphen muss so über das assoziierte Informationsobjekt erfolgen, wodurch sich die Synchronität zwischen beiden Strukturen automatisch ergibt.

Ein weiteres potentielles Problem bei der Verwendung zusätzlicher Datenstrukturen ist die externe Synchronität der Datenmodelle. Änderungen an den eigentlichen Präsentationsdaten müssen auch in den Informationsobjekten nachgezogen werden. Da Referenzen auf Präsentationsdaten in den Informationsobjekten als Hyperlink gespeichert werden, sind auch für das AR System die jeweils aktuellsten Versionen der Dokumente verfügbar. Lediglich das Löschen oder Verschieben von Dokumenten muss in den Informationsobjekten nachgezogen werden.

Die Problematik der Aktualität von Informationsobjekten kann gänzlich umgangen werden, wenn diese automatisch zur Laufzeit generiert werden. Zu einem Ort oder einem Objekt werden hierbei alle Dokumente gesucht, die auf das entsprechende Objekt verweisen. Hierfür ist es allerdings notwendig, dass die Informationen bereits bezüglich Struktur, Granularität und auch hinsichtlich sinnvoller Benennung den notwendigen Kriterien entsprechen.

Die Informationsobjekte werden in einer hierarchischen Baumstruktur gehalten, die vom eigentlichen Render-Szenengraphen unabhängig ist, was es wiederum erlaubt die Modellierung der Informationen losgelöst von der Detaillierungsstufe der Render-Szene vorzunehmen. Zusätzlich ermöglicht diese Unabhängigkeit von Render-Geometrien auch die Erstellung von Informationsräumen für Orte, zu denen kein detailliertes Geometrie-Modell vorhanden ist.

Jedes sichtbare Informationsobjekt wird Teil der graphischen Benutzeroberfläche und ist als solcher selektierbar. Alle Informationsobjekte, unabhängig ob sie mit einer Position oder einem Objekt verknüpft sind, müssen somit für den Anwender auch als Informationsobjekt erkennbar sein. Das Vorgehen für die beiden Typen von Informationsobjekten ist dabei getrennt zu betrachten. Ist ein Informationsobjekt mit einem Ort verknüpft, muss an dieser Position eine entsprechende Geometrie im Szenengraphen erzeugt werden. Die graphische Repräsentation sollte in diesem Fall möglichst blickrichtungsunabhängig sein, so dass ein Informationsobjekt von unterschiedlichen Positionen gleich gut wahrgenommen werden kann. Es bieten sich hierfür einfache Primitive wie Kugel oder Würfel ebenso an, wie Billboards, texturierte Polygone die sich immer in Richtung des Betrachters ausrichten.

Da der Anwender über die Informationsobjekte auf Daten zugreifen kann, ist es unbedingt notwendig ihm eine adäquate optische Darstellung zur Verfügung zu stellen. Hauptaufgabe ist dabei natürlich der Hinweis darauf, dass zu einem Ort oder Objekt überhaupt Informationen verfügbar sind. Zusätzlich sollte erkennbar sein um welche Art von Informationen es sich dabei handelt. Wobei diese Kategorisierung von Informationen meist applikationsspezifisch vorzunehmen ist. Der Servicetechniker kann so schnell erfassen, ob für ein Aggregat eine Ein- und Ausbauanleitung vorliegt oder nur eine Stückliste der Einzelteile. Der Besucher einer antiken Stätte will hingegen erkennen, ob zu einem Gebäude Informationen zum Thema Architektur vorliegen.

Neben der Sichtbarkeit der Informationsobjekte für den Benutzer ist es natürlich wichtig, dass der Benutzer auch darauf zugreifen kann. Besonders bei komplexen technischen Anlagen liegen Bedienelemente oft eng beieinander, was zu einer entsprechend hohen Dichte an Informationsobjekten führt. Abhilfe können hier Mechanismen zur Informationsfilterung schaffen, die abhängig vom Anwender-

profil und der aktuellen Aufgabe oder Arbeitssituation die darzustellende Informationsmenge effizient steuern.

4.2.6 Informationsfilterung

Um zu verhindern, dass der Benutzer mit einer übermäßigen Menge an Informationen konfrontiert wird, sind besonders in mobilen Systemen effiziente Mechanismen bereitzustellen, die eine Filterung der darzustellenden Informationen vornehmen. Benford und Fahlén [BENF93] beschreiben ein allgemeines räumliches Interaktionsmodell, bei dem das Bewusstsein zweier Objekte füreinander in einem bestimmten Medium von deren *Fokus* und *Nimbus* abhängig sind. Der Fokus beschreibt in diesem Zusammenhang die Menge an Aufmerksamkeit für ein anderes Objekt, während der Nimbus den Grad an Aufmerksamkeit beschreibt, den ein Objekt verursacht. In [JULI01] wird dieser Mechanismus in einem mobilen Augmented Reality System angewandt, um die Sichtbarkeit von virtuellen Objekten zu steuern. Jedes Objekt, das potentiell angezeigt werden kann, besitzt einen Vektor in dem für unterschiedliche Aufgaben ein Relevanzwert eingetragen ist. Demgegenüber steht ein Vektor, der die aufgabenbezogenen Ziele des Benutzers beschreibt. Die Sichtbarkeit wird hier aus der Kombination der beiden Vektoren bestimmt.

Das hier vorgestellte Verfahren erweitert dieses Modell um einige für das Anwendungsgebiet bedeutende Funktionalitäten. Die Filterfunktionalität arbeitet dabei auf dem erweiterten Weltmodell und den darin gespeicherten Informationsobjekten und deren Attributen. Die Speicherung von Attributen, sowohl zu den Informationsobjekten selbst, als auch zu den darin verlinkten Dokumenten, erlaubt es, eine effektive Steuerung der Sichtbarkeit von Informationen vorzunehmen.

Die Ermittlung der Sichtbarkeit erfolgt dabei hauptsächlich auf der Analyse der Blickrichtung und auf dem Abstand zum Objekt, mit dem die Informationen verlinkt sind. Das Objekt selbst besitzt spezielle Attribute, wie die Richtung und den Abstand aus dem es sichtbar ist.

Klassische Level-Of-Detail (LOD) Techniken basieren allein auf dem Abstand eines Objekts vom Betrachter und steuern basierend auf dieser Information die Detaillierung des darzustellenden Objekts. Ein ähnliches Verfahren wird hier für die Filterung der Informationen verwendet. Bezogen auf die Nomenklatur von *Fokus* und *Nimbus* bedeutet dies, dass bei typischen LODs der *Nimbus* kreisrund ist, mit dem jeweiligen Objekt im Zentrum des Kreises.

Da Informationen meist nicht aus allen Richtungen gleichermaßen sichtbar sein sollten, kann zusätzlich zur Entfernung des Objektes auch die Richtung ausgewertet werden, aus der der Anwender auf das Objekt schaut. So können beispielsweise Informationen, die Bedienelemente auf der Rückseite einer Maschine beschreiben, bei der Wahrnehmung der eigentlich relevanten Informationen zu Objekten auf der Vorderseite der Maschine stören. Eine Filterung, die nur über die Distanz zu Informationen arbeitet, kann hier nicht immer die gewünschten Ergebnisse liefern, da die Informationen auf der Rückseite der Maschine ebenfalls relativ nah zum Anwender sein können. Abhilfe schafft ein kegelförmiger *Nimbus*, der als Achse die Hauptsichtbarkeitsrichtung eines Objekts erhält. Diese Achse erhält eine ähnliche Funktion wie die Normale eines Polygons beim Rendering. Die Normale steuert sowohl Sichtbarkeit als auch die Darstellung des Polygons über dessen Materialeigenschaften und die

Beleuchtungssituation. Polygone, deren Normale vom Betrachter weg weisen, werden bei geeigneten Render-Einstellungen nicht dargestellt.

Ähnlich der Normalen eines Polygons erhält ein Informationsobjekt einen Vektor, der die Richtung angibt, aus der ein Objekt sichtbar ist. Zusammen mit einem Winkel, der den Bereich begrenzt, in dem das Objekt wahrnehmbar ist, ergibt sich so ein Sichtbarkeitskegel.

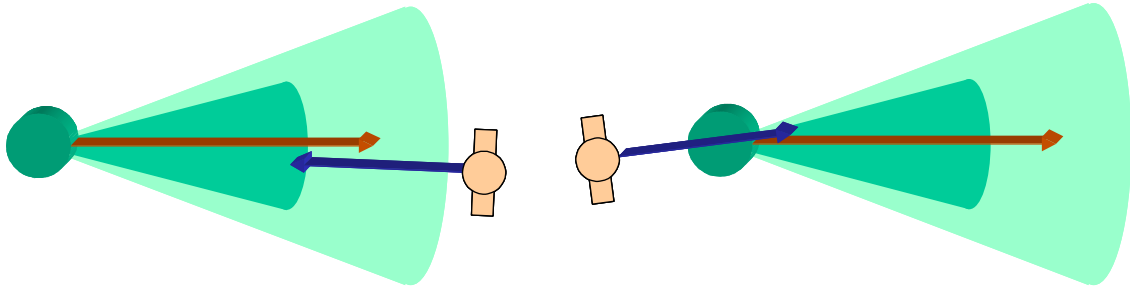


Abbildung 32: Konfigurationen, bei denen die Sichtbarkeitsbedingungen nicht erfüllt sind

Entsprechend der Kombination der beiden Kriterien Abstand zum Objekt und Richtung aus der ein Anwender das Objekt sieht, ergeben sich unterschiedliche Sichtbarkeitskonfigurationen. Unabhängig vom Abstand des Betrachters ist ein Objekt nur sichtbar, wenn sich der Betrachter aus der Richtung des Sichtbarkeitskegels nähert. Allerdings muss auch das zweite Kriterium, der Abstand zum Objekt unterschreitet einen Schwellwert, erfüllt sein, damit ein Objekt wirklich sichtbar wird. Unterschiedliche Konfigurationen des Sichtbarkeitstests können Abbildung 32 und Abbildung 33 entnommen werden.

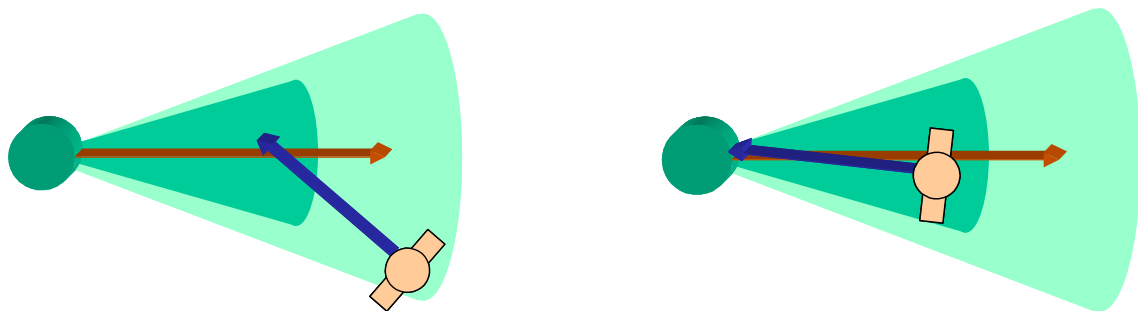


Abbildung 33: Erfüllte Sichtbarkeitskriterien und unterschiedliche LOD-Stufen

Ein Objekt ist genau dann sichtbar, wenn beide Bedingungen erfüllt sind. Erst wird ein Test durchgeführt, ob der Winkel zwischen Normalenvektor und Blickrichtung klein genug ist. Verläuft dieser Test erfolgreich, wird anschließend getestet, ob auch die Abstandsbedingung erfüllt ist.

Der Testalgorithmus sieht in Pseudocode folgendermaßen aus:

```
for ( all objects ){
    if ( (angle - normal) < testAngle ){
        if( distance < threshold ){
            showObject();
        }
    }
}
```

Dieser recht einfache Test arbeitet bei der Verringerung der angezeigten Informationsmenge bereits sehr effizient. Allerdings ist das Ergebnis stark von der Modellierung der Umgebung abhängig und es lässt sich so einzig die Sichtbarkeit auf Objektebene steuern. Objekte tauchen relativ unvermittelt im Sichtfeld des Benutzers auf und verschwinden auch wieder ebenso unvermittelt. Damit der Benutzer die Sichtbarkeit bewusster steuern kann, sind eine Erweiterung des Algorithmus und zusätzliche Attribute notwendig. Ähnlich der Level Of Detail Technik in der Computergrafik, lassen sich verschiedene grafische Repräsentationen der Datenobjekte umschalten. Auf der niedrigsten Stufe bekommt der Benutzer einzig den Hinweis auf die Existenz eines Informationsobjekts präsentiert. Mit abnehmender Entfernung können weitere Details, wie die Bezeichnung eines Bedienelements als Tooltip dargestellt werden.

Besonders in Umgebungen mit hoher Informationsdichte, also beispielsweise vor einer komplexen technischen Anlage, ist der reine Ortsbezug als Filterkriterium meist nicht ausreichend. In solchen Fällen kann es leicht zu so genanntem display-cluttering, also der Überladung des Displays mit virtuellen Informationen kommen. Um dies zu verhindern, müssen zusätzliche Parameter für die Filterung verwendet werden. Zusammen sollen diese Parameter die Situation, in der sich der Benutzer aktuell befindet, adäquat abbilden. Das resultierende Modell wird auch als der Kontext der Anwendung bezeichnet. Je besser dieses Modell den tatsächlichen Anforderungen angepasst ist, umso präziser kann die Menge der Informationen bestimmt werden, die ein Benutzer aktuell benötigt.

Der aktuelle Kontext kann wiederum als Vektor von Attributen beschrieben werden. Sinnvolle Einträge können beispielsweise die aktuelle Aufgabe oder besondere Präferenzen des Benutzers sein. Die Einträge dieses Vektors können nun elementweise mit den Attributen der Informationsobjekte verglichen werden, wobei eine Gewichtung der einzelnen Einträge möglich ist. Das Ergebnis dieses Vergleichs ist ein skalarer Wert, der wiederum mit einem Relevanz-Schwellwert abgeglichen werden kann.

```
for(i=0; i<numAttribute;i++)
{
    kontextValue = kontext[Attribute[i].name].value();
    relevanz += Attribute[i].weight *
                ( Attribute[i].value() && kontextValue);
}
if ( relevanz > threshold )
    showObject();
```

Das Ergebnis dieser zweiten Stufe wird auf die Objekte angewendet, die den ersten Sichtbarkeittest bestanden haben und kann einen Teile dieser Objekte, die aktuell für den Anwender nicht relevant erscheinen, wieder deaktivieren. Das Ergebnis ist eine Menge an Informationsobjekte, die sowohl die Normalen- und LOD-Tests als auch diesen Relevanz-Test bestanden haben.

Da diese Tests auf dem Informationsraum arbeiten, der im erweiterten Weltmodell gespeichert ist, müssen sich vorher bereits alle in Frage kommenden Informationsobjekte auf dem mobilen Rechner befinden. Für viele Anwendungen bedeutet dies, dass die Informationsobjekte über eine Netzwerkverbindung übertragen wurden, ohne dass sie jemals angezeigt werden, da sie keine Relevanz für die aktuelle Aufgabe des Anwenders besitzen. Zur Reduzierung des Datenaufkommens auf dieser Datenverbindung ist es sinnvoll, hier nur so genannte Proxy-Objekte zu übertragen, die lediglich die für die Tests benötigten Informationen beinhalten. Weitere Informationen werden erst später, bei Bedarf übertragen. Die Informationsfilterung stellt sich somit auch als ein Kandidat für die Verteilung auf andere Rechner dar, im speziellen den Server der Informationsobjekte dar. In diesem Fall werden kontinuierlich die Position und Blickrichtung des Benutzers über das Netz an die Filterkomponente übermittelt. Informationsobjekte und deren Geometrie zur Darstellung werden nur im Falle, dass alle oben beschriebenen Tests erfolgreich durchgeführt wurden zum mobilen Rechner übertragen. Denkbar ist hier aber auch eine Aufspaltung der Filterfunktionalität in mehrere hintereinander zu durchlaufende Filterstufen, die je nach Bedarf und Last auf den Server verlagert werden können. Hieraus ergibt sich ein mehrstufiges Filtersystem, das neben der Reduzierung der Netzlast auch eine verringerte Rechenlast bei der clientseitigen Filterung erlaubt. Während der Relevanztest vollständig serverseitig durchgeführt werden kann, ist der Test auf die Sichtbarkeitskriterien nur mit entsprechend erhöhten Schwellwerten durchzuführen.

4.3 Geräteschnittstelle

Ein charakteristisches Feature von MR Systemen ist eine Geräteabstraktionsschicht, die es ermöglicht die verschiedensten Ein- und Ausgabegeräten und auch Tracking-Systemen zu verwenden. Die Geräte werden dafür in abstrakte Gruppen aufgeteilt, die üblicherweise nach dem Typ der Daten, die sie generieren, unterschieden werden. [FELGER95] Moderne Geräteschnittstellen bieten darüber hinaus weitere Funktionalitäten, die auch das Ansprechen von Tracking- und Interaktionsgeräten erlauben, die im Netzwerk verteilt an unterschiedliche Rechner angeschlossen sind. Im Folgenden werden deshalb die

relevanten Entwicklungen auf diesem Gebiet betrachtet, bevor detailliert auf das speziell für die Realisierung kombinierter Tracking-Verfahren in mobilen AR Systemen abgestimmte Tracking-Framework eingegangen wird.

4.3.1 Bestehende Systeme

4.3.1.1 OpenTracker

Das OpenTracker Framework [REIT01b] ist unter der LGPL Lizenz als OpenSource verfügbar und basiert auf einem Datenflusskonzept für multi-modale Events. Das Datenflusskonzept wird durch einen gerichteten Graphen realisiert, der über eine XML-Beschreibung konfiguriert wird.

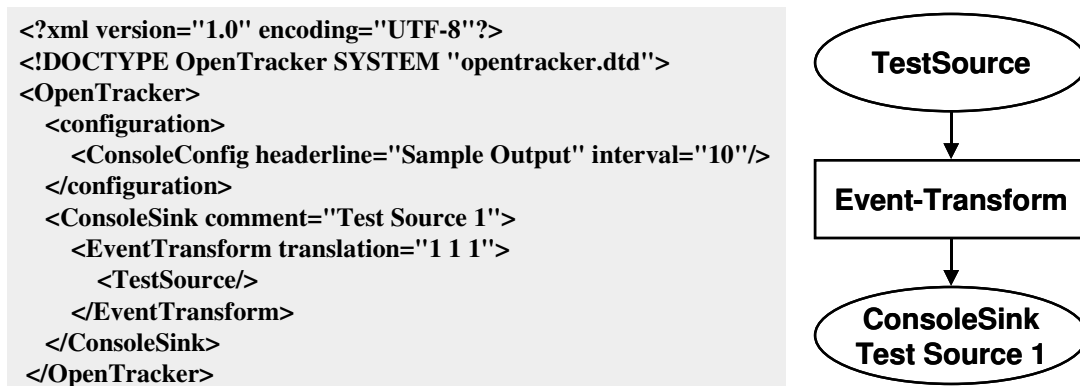


Abbildung 34: XML-Beschreibung und korrespondierender Datenflussgraph [OTUG03]

Es existieren drei Klassen von Knoten, Source-Knoten, Filter-Knoten und Sink-Knoten. Source-Knoten kapseln die Gerätetreiber und bilden somit die Geräteabstraktion. Filter-Knoten beinhalten generische Operationen, aus denen die benötigten Datentransformationen und –manipulationen aufgebaut werden. Ausgegeben werden die Daten über so genannte Sink-Knoten.

Die Anbindung netzwerkweiter Tracking- und Interaktionsgeräte erfolgt über spezielle Source- und Sink-Knoten und ein eigenes Netzwerkprotokoll.

4.3.1.2 IDEAL

Die Designziele bei der Entwicklung der Geräteschnittstelle IDEAL (Interaction DEvice Abstraction Layer) [FRÖHL00] sind der schnelle und verlässliche Zugriff auf netzwerkweit verteilte Interaktions- und Tracking-Geräte, die flexible und transparente Verwendung einer großen Zahl solcher Geräte in einer Anwendung sowie die einfache Benutzung und Wartung des Systems auch durch nicht VR-Experten.

Die Applikation greift auf eine logische Geräteschnittstelle zu, die eine Klassenhierarchie logischer Geräte zur Verfügung stellt. Die Gerätetreiber werden in so genannten Servern gekapselt, Prozessen, die sowohl die Schnittstelle überwachen als auch die Kommunikation mit der Geräteschnittstelle bewältigen. Zu jedem Gerätetyp muss ein eigener, gerätespezifischer Server geschrieben werden. Auf

jedem Rechner, an dem Geräte angeschlossen sind, läuft zusätzlich ein Demon, der bei Bedarf die Server startet und verwaltet.

Über IDEAL angebotene logische Geräte können über einen symbolischen Namen angesprochen werden, der sie eindeutig identifiziert.

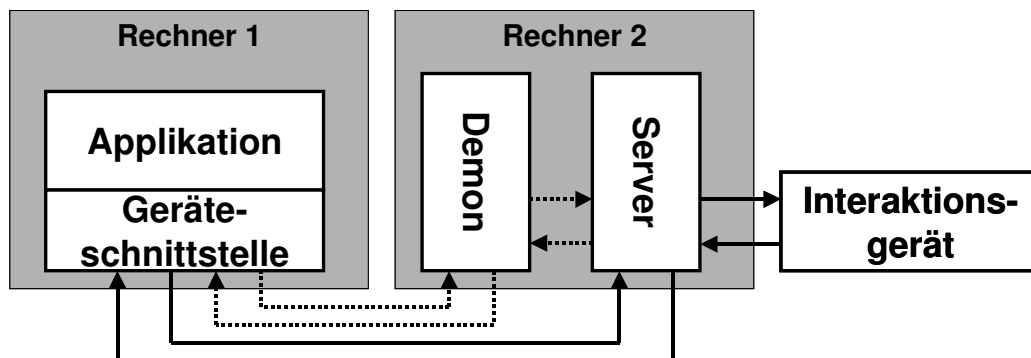


Abbildung 35: Netzwerkweite Geräteanbindung über IDEAL

Die Konfiguration der IDEAL Geräteschnittstelle und der angebundenen Geräte erfolgt über Dateien, die jeweils zum Start der Geräteschnittstelle und der einzelnen Server ausgewertet werden.

Speziell in mobilen Applikationen, bei denen alle Geräte an den lokalen Rechner angeschlossen werden, bedeutet das Konstrukt aus Servern, Demons und der Geräteschnittstelle einen unnötigen Overhead und zusätzlichen Ressourcenverbrauch. Die in den Servern eingebettete Geräteabstraktion und die Netzwerkfunktionalitäten werden getrennt. Die so entstandenen Low Level Device Interfaces (LLDI) können direkt an die Geräteschnittstelle angebunden werden.

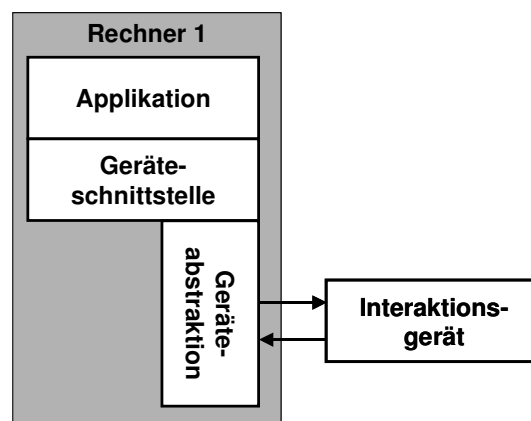


Abbildung 36: Direkte Geräteanbindung über IDEAL-LLDI

Die Konfiguration der Geräte kann entweder über Dateien oder Parametercontainer erfolgen, die von der Applikation gefüllt werden. Dies ermöglicht auch zur Laufzeit Änderungen der Tracker-Konfiguration vorzunehmen.

4.3.2 Hybride Tracking-Verfahren

Die Geräte-Schnittstellen OpenTracker (siehe Abschnitt 4.3.1.1) und IDEAL (4.3.1.2) verfolgen das Ziel, eine möglichst allgemeine Unterstützung für die Anbindung verschiedenster Tracking- und Interaktionsgeräte zu bieten. Verschiedene Geräte oder Verfahren werden grundsätzlich auf Basis der Geräte-Abstraktion gleich angesprochen und können so einfach gegen Geräte der gleichen Klasse ausgetauscht werden. Konzepte zur Realisierung von hybriden Tracking-Verfahren sind in IDEAL überhaupt nicht und in OpenTracker nur rudimentär vorhanden. Während in IDEAL über die IDs der verwendeten abstrakten Geräte auf diese zugegriffen und somit mehrere Geräte parallel angesprochen werden können, erfolgt dieser Zugriff auf Geräte-Daten in OpenTracker über spezielle Knoten, Sinks (Senken). In diese Sinks können die Daten mehrerer anderer Knoten zusammenlaufen und somit die Kombination verschiedener Verfahren realisiert werden.

Wie bereits in Abschnitt 2.2.1.5 angesprochen, ist aus heutiger Sicht für die Umsetzung mobiler AR Anwendungen die Verwendung von visionbasierten Tracking-Verfahren und deren Unterstützung durch andere Verfahren unerlässlich. Nur so lässt sich ein Tracking-System aufbauen, dass mit vertretbarem Hardwareaufwand auch einen größeren Arbeitsbereich abdecken kann. Allerdings sind dabei verschiedene Nebenbedingungen zu beachten:

- Modellbasierte Tracking-Verfahren benötigen ein Geometrie-Modell der Umgebung, das mit dem Kamerabild verglichen wird. Ein solches Modell sollte idealer Weise den gesamten Arbeitsbereich des Anwenders abdecken können, so dass jederzeit gültige Tracking-Ergebnisse generiert werden können. Zusätzlich sollte das Modell auch bei der Annäherung, wenn nur ein kleiner Ausschnitt davon sichtbar ist, noch genug Merkmale aufweisen, die mit den gefundenen Features im Bild abgeglichen werden können.
- Rein featurebasierte Verfahren, die von Bild zu Bild arbeiten, benötigen eine Initialisierung, damit sie die für AR Anwendungen notwendige Transformation ihrer relativen Posen in ein definiertes Referenz-Koordinatensystem vornehmen können.

Kategorisiert werden kann diese Unterstützung nach den Phasen des Tracking-Vorgangs, in denen sie notwendig ist. Wir unterscheiden hier in Initialisierung, Tracking und Re-Initialisierung. Die Phase der Initialisierung sollte dabei idealerweise nur einmal durchlaufen werden, um die initiale Relation zwischen Benutzer und Umgebung zu bestimmen. Je nach verwendetem Verfahren kann dieser Vorgang deutlich länger dauern als das eigentliche Tracking, so dass ein wiederholtes Eintreten in diese Phase eine störende Unterbrechung darstellt.

Wurde die Phase der Initialisierung erfolgreich durchlaufen, so tritt das System in die eigentliche Tracking Phase ein. Hier können sehr performante Verfahren eingesetzt werden, die lediglich von Frame zu Frame tracken und daraus die relative Pose-Veränderung bestimmen. Verlieren solche Verfahren einmal den Track, sei es durch Verdeckungen im Bild, durch starke Bewegung verwischte Bilder oder auch durch zu große Unterschiede zwischen zwei aufeinander folgenden Frames, besitzen sie meist nicht die Fähigkeit aus eigener Kraft wieder aufzusetzen. Sie müssen also das Versagen erkennen können, um dann zu signalisieren, dass das Tracking unterbrochen wurde. Das System geht also in die Re-Initialisierung über, bei der typischerweise ein Verfahren zum Einsatz kommt, dass vom Rechenaufwand zwischen der Initialisierungs- und der Tracking-Lösung liegt. Solche Verfahren benutzen

Informationen zum Wiederaufsetzen, die während dem erfolgreichen Tracken der Szene gesammelt wurden. Hieraus lässt sich entnehmen, dass auch während des Trackings eine Teilkomponente der Re-Initialisierung aktiv ist, die zumindest sporadisch Daten aufnimmt, die für das Wiederaufsetzen in der eigentlichen Re-Initialisierungsphase als Referenzdaten genutzt werden können.

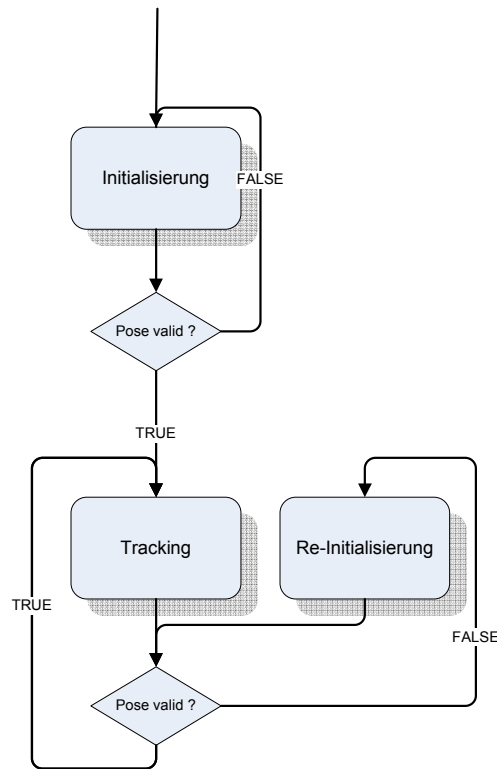


Abbildung 37: Übergänge zwischen den verschiedenen Phasen des visionbasierten Trackings

Gültige Kamera-Posen liegen somit nach erfolgreicher Initialisierung, während des Trackings und bei erfolgreicher Re-Initialisierung vor. Um auch während der Re-Initialisierung gültige Posen liefern zu können und auch für die Kompensation der aufakkumulierten Fehler der Frame zu Frame Tracking-Verfahren sind zusätzliche Verfahren und Strategien notwendig, wie sie im nachfolgend beschriebenen Tracking-Framework realisiert wurden.

4.3.3 Tracking-Framework

Ziel des hier beschriebenen Tracking-Frameworks ist der einfache Aufbau robuster und performanter Tracking-Systeme, die aus mehreren unterschiedlichen Tracking-Verfahren und –Technologien aufgebaut werden. Hauptaugenmerk liegt dabei auf dem schnellen Zusammenfügen der Einzelverfahren und der einfachen Parametrierbarkeit sowohl zum Start des Frameworks als auch zur Laufzeit. Entwickeln von Tracking-Verfahren und Integratoren von Gesamt-Lösungen soll somit eine Basis an die Hand gegeben werden, die es ermöglicht, sowohl unter Laborbedingungen als auch in realen Anwendungsszenarien möglichst schnell die optimale Lösung für einzelne spezielle Anwendungsfälle zu finden. Hintergrund ist, dass aus heutiger Sicht kein einzelnes Tracking-Verfahren so universell einsetzbar ist,

dass es unter sehr unterschiedlichen Bedingungen konstant gute Ergebnisse liefern kann. Vor allem bei zusammengesetzten Tracking-Lösungen ist die optimale Wahl der Einzelverfahren und deren Abstimmung eine Aufgabe, die noch sehr viel Erfahrung und hohen Test- und Evaluierungsaufwand erfordern.

Wie aber bereits im vorherigen Abschnitt angedeutet, erfordern die schnellen visionbasierten Tracking-Verfahren, die ohne Marker und sonstige Instrumentierung arbeiten, die Unterstützung anderer Verfahren für Initialisierung und Re-Initialisierung. Der Verlust des Trackings liegt dabei am starken Gebrauch von Kohärenzen zwischen zwei aufeinander folgenden Videobildern der verwendeten Algorithmen. Nur so ist es möglich Merkmale stabil und mit sehr hoher Framerate über eine Sequenz von Bildern zu verfolgen. Kann keine ausreichende Anzahl stabiler Features verfolgt werden, sei es durch zu starke Kamerabewegungen oder Verdeckungen, kann keine Bestimmung der Kamera-Pose erfolgen. Eine (Re)-Initialisierung des Verfahrens wird notwendig. Diese Initialisierung wird üblicherweise über ein zweites Tracking-Verfahren realisiert, das eine gültige erste Lösung liefert, auf die das featurebasierte Verfahren wieder aufsetzen kann (Siehe Abbildung 38).

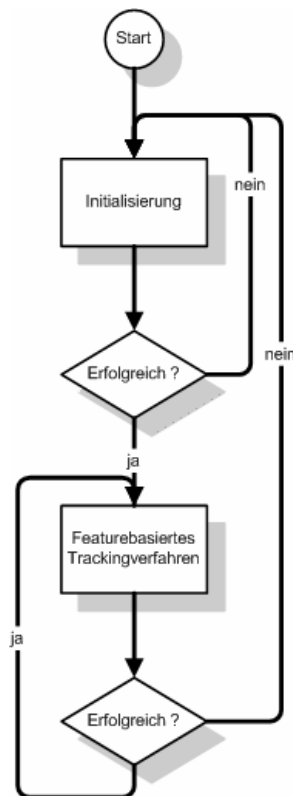


Abbildung 38: Ablauf des kombinierten Tracking-Verfahrens ohne Beschleunigungssensor

Die Anforderungen an das Tracking-Verfahren zur Initialisierung sind, dass dies in der Lage sein muss, sowohl absolute Positions- als auch Orientierungsdaten zu liefern. In weitläufigen Outdoor Anwendungen kann die (Re)-Initialisierung über die Kombination eines GPS Empfängers, der die Positionsinformation liefert, und einem digitalen Kompass bewerkstelligt werden. Bei Anwendungen in Gebäuden dagegen bietet sich die Verwendung von markerbasiertem Tracking an, da eine Kamera bereits mitgeführt wird. Meist reicht hier das Anbringen eines Markers außerhalb des Arbeitsbereichs, der für die (Re)-Initialisierung in das Blickfeld der Kamera gebracht werden muss.

Trotz Re-Initialisierung verliert das System allerdings in jedem Fall die Orientierung und kann bis zum erfolgreichen Abschluss der Re-Initialisierung keine gültige Kamera-Pose liefern. Eine Strategie, um dies zu verhindern ist der zusätzliche Einsatz von Beschleunigungssensoren, die während des Ausfalls des Hauptsystems zumindest Orientierungsdaten liefern. Somit ist es unter der Annahme, dass der Benutzer nur Kopfdrehungen durchführt, auch weiterhin möglich, basierend auf der letzten gültigen 6D Pose ausreichend genaue Daten für eine kontinuierliche Registrierung zu erhalten.

Das oben beschriebene Verfahren kann als Beispiel für unzählige Möglichkeiten der Realisierung kombinierter Tracking-Verfahren angesehen werden. Schon mit den Varianten GPS und Marker-Tracking stehen zwei komplementäre Verfahren bereit, die sehr unterschiedlicher Einsatzbereiche abdecken. Wenn verfügbar, können weitere Systeme, wie WLAN-Peilung oder RFIDs als zusätzliche Quellen für Positions- oder Orientierungs-Daten verwendet werden.

4.3.3.1 Framework-Architektur

Um nun in einem mobilen AR System diese Variabilität der Tracking-Lösung abbilden zu können, ist ein Rahmensystem notwendig, das die einfache Komposition komplexer Systeme aus mehreren Tracking-Subsystemen erlaubt. Trotz der geforderten Flexibilität ist allerdings die effiziente Gestaltung der Kommunikation und des Datenflusses zwischen den Subsystemen mit höchster Priorität zu bewerten.

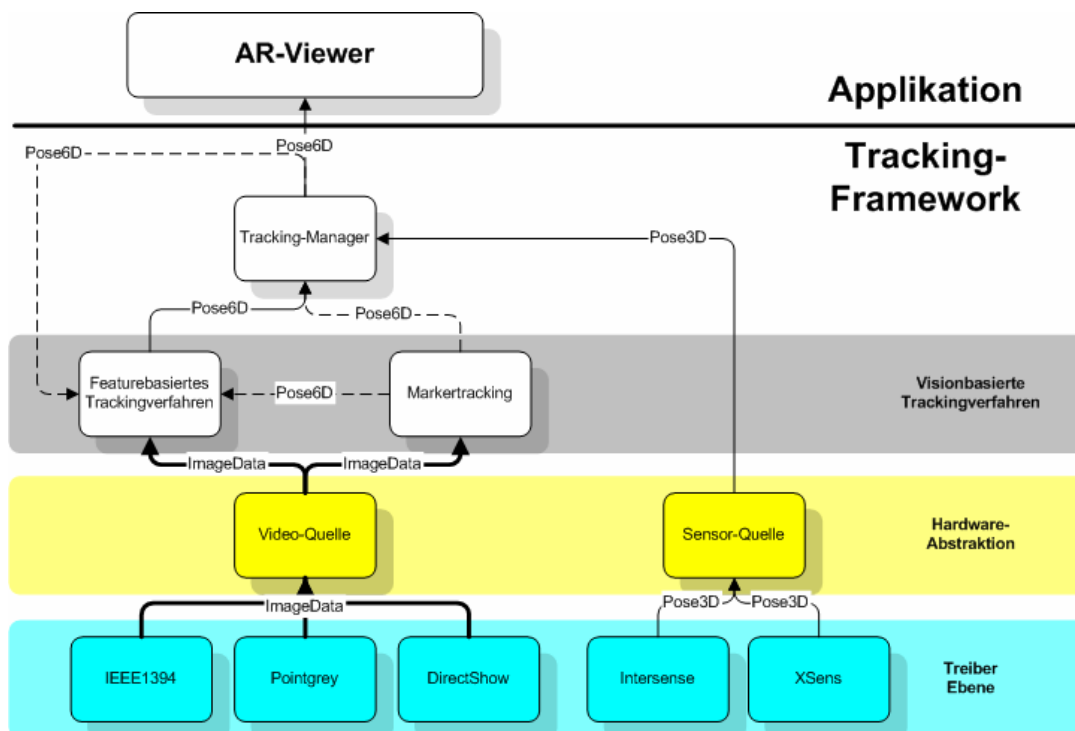


Abbildung 39: Datenflüsse im Tracking-Framework

Die Komponenten des Tracking-Frameworks können in drei Ebenen eingeteilt werden, die sich hauptsächlich hinsichtlich der Nähe zur Hardware Ebene unterscheiden. Abbildung 39 zeigt eine beispielhafte Konfiguration des Frameworks, bei der die Initialisierung entweder über das letzte integrierte

Ergebnis des Tracking-Frameworks erfolgt, oder bei Misserfolg über ein markerbasiertes Tracking-Verfahren.

4.3.3.2 Komponenten

Die einzelnen Tracking-Verfahren sind jeweils als eigenständige Komponente realisiert, die anderen Komponenten einen bestimmten Dienst zur Verfügung stellt. Im Falle eines visionbasierten Tracking-Verfahrens ist dies die Bestimmung der aktuellen Kamera-Pose zu einem Video-Bild, das als Eingabe dient. Die Tracking-Komponente muss also auf eine andere Komponente zugreifen, die das Video-Bild bereitstellt. Komponenten stellen Dienste zur Verfügung und benötigen wiederum die Dienste anderer Komponenten, um überhaupt arbeiten zu können. Die Schnittstelle einer Komponente zu ihren Diensten und ihren Bedürfnissen ist über Slots, ähnlich den Feldern eines Szenengraphknotens realisiert. Allerdings speichern diese Slots nicht selbst die Daten, sondern verweisen auf Datenobjekte in einem gemeinsamen Daten-Pool. Schreibt eine Komponente in einen Ausgangs-Slot, so schreibt sie in das assoziierte Daten-Objekt (Abbildung 40).

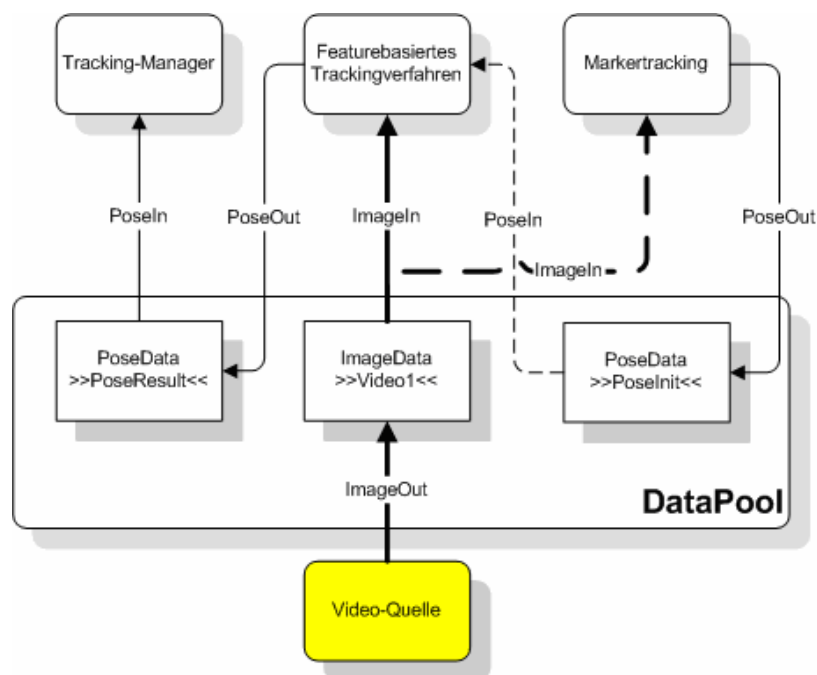


Abbildung 40: Datenaustausch über den gemeinsamen Daten-Pool

Andere Komponenten, die mit diesem Daten-Objekt als Eingang assoziiert sind, werden über die Änderung benachrichtigt und können lesend darauf zugreifen. Da Datenobjekte nur einmal im Daten-Pool angelegt werden und mehrere Komponenten darauf Zugriff besitzen, kann unnötiges Kopieren von Daten vermieden werden, was besonders bei großen Datenmengen, wie Video-Bildern ins Gewicht fällt.

Daten-Objekte werden über ihren Namen identifiziert, so dass keine festen Verbindungen zwischen Komponenten und Datenobjekten bestehen müssen. So können auch zur Laufzeit Änderungen an der Konfiguration des Frameworks vorgenommen werden. Eine Komponente benötigt somit kein Wissen,

woher der Inhalt eines Datenobjekts stammt, aus dem es liest oder wer der Empfänger der Daten ist, die eine Komponente hinein schreibt. Das Datenobjekt muss lediglich dem Typ des Slots entsprechen. So ist es beispielsweise möglich die Initialisierung über Marker durch ein ähnlich verlässliches Verfahren zur Laufzeit zu ersetzen, wenn dies die Situation erfordert.

4.3.3.3 Multithreading

Da jede Komponente in einem unabhängigen Thread ablaufen kann, ist die Synchronisierung der Zugriffe auf gemeinsam genutzte Daten notwendig. Alle Zugriffe auf Datenobjekte sind zu diesem Zweck über Synchronisationsprimitive, hier werden Locks verwendet, gesichert. Jeder Zugriff auf ein solches Objekt wird durch eine `lock()` und `unlock()` Folge eingerahmt, die weitere zeitgleiche Zugriffe auf dieses Objekt verhindert (Abbildung 41). Jedes Datenobjekt besitzt sein eigenes Lock, so dass hier die feinst mögliche Granularität der Synchronisierung erreicht wird, um die Wartezeiten an einem Lock zu minimieren.

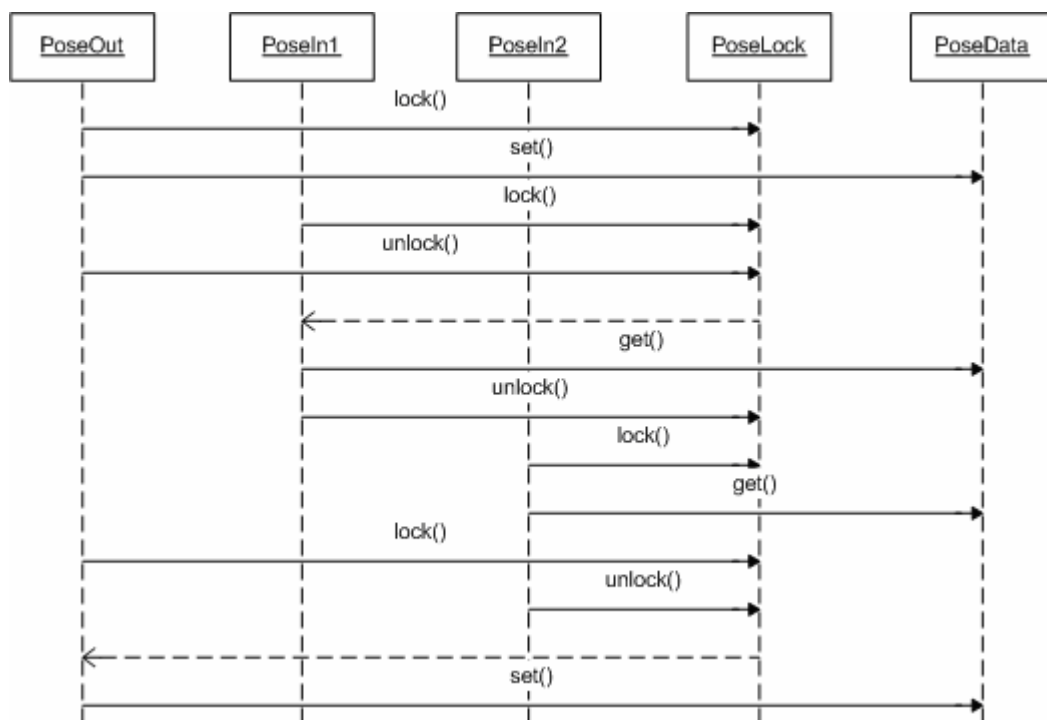


Abbildung 41: Synchronisierung über Datenobjekte

Das Multithreading ermöglicht es, den Ablauf der Applikation und der einzelnen Tracking-Verfahren bestmöglich voneinander zu entkoppeln. Eine Anfrage nach der aktuellen Pose kann beantwortet werden, ohne eine neue Berechnung durchzuführen. Das jeweilige Tracking-Verfahren liefert sobald ein aktuelles Datum vorliegt, dies beim Tracking-Manager ab.

4.3.3.4 Immediate-Methoden

Obwohl im Daten-Pool jeweils die aktuellsten Werte stehen, die eine Komponente geliefert hat, kann dieser Wert bereits veraltet sein. Dies liegt vor allem daran, dass der Datenaustausch über den Data-Pool nur einen Push-Betrieb erlaubt. Ist allerdings gefordert, dass der Eingabewert eine zugesicherte

Aktualität besitzt, ist ein zusätzlicher Mechanismus notwendig, der einen Pull-Betrieb ermöglicht. Fragt eine Komponente eine andere nach der aktuellen Pose, so wird hierdurch eine Neuberechnung der Pose angestoßen. Auf diesem Weg können die mit hoher Updaterate vorliegenden Rotations-Daten eines Inertial-Sensors zeitnah abgeholt werden. Der Zugriff auf eine Komponente erfolgt dabei über eine zusätzliche Methode des Komponenten-Interface, so dass auch der Speicher-Footprint einer Komponente nicht wesentlich vergrößert wird. Die Identifikation der Zielkomponente erfolgt wiederum über deren eindeutige Id.

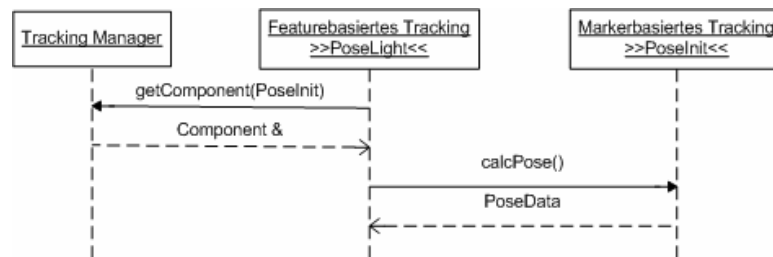


Abbildung 42: Direkter Zugriff auf andere Komponente

Ebenso, wie der Datenaustausch über Slots und den Daten-Pool ermöglicht auch diese Methode maximale Transparenz und minimales Wissen einer Komponente über den Aufbau des Frameworks und andere Komponenten. Über die Kombination der beiden Wege Daten auszutauschen können Tracking-Systeme mit maximaler Flexibilität und Performanz zusammengestellt werden. Zusätzlich kann der kombinierte Einsatz beider Verfahren auch zu erhöhter Robustheit gegen Fehlkonfigurationen dienen, wenn beim Misserfolg einer direkten Verbindung ein passendes Datenobjekt zum Verbinden über den Datenpool gesucht wird.

Neben den Verbindungen der Komponenten und den Datenfluss im Netzwerk ist auch die Konfiguration der einzelnen Komponenten zu beachten. Alle Komponenten des Frameworks besitzen einen spezifischen Satz von Attributen, die ihr Verhalten steuern. Diese Attribute können sowohl über XML-Konfigurationsdateien eingelesen werden, als auch zur Laufzeit von der Applikation gesetzt werden.

4.3.3.5 Integration in den AR-Browser

Der AR-Browser spricht das Tracking-Framework über ein abstraktes `DeviceInterface` an, so dass mehrere Tracking-Frameworks transparent parallel oder auch andere Interfaces, wie `OpenTracker` in den AR-Browser über diese Schnittstelle eingebunden werden können.

Das realisierte System beinhaltet zwei bildbasierte, markerlose Tracking-Verfahren und einen Inertial-Sensor. Die eigentliche Ergebnispose wird von einer zusätzlichen Komponente `SensorFusion` berechnet. Eines der bildbasierten Tracking-Verfahren arbeitet mit einer perspektivischen Kamera, während das andere eine 180° Fischaugenkamera verwendet (Abbildung 43). Da zwei bildverarbeitende Verfahren zu viel Prozessorlast auf einem Rechner verursachen, um angemessene Frameraten zu erzielen, wurde das auf der Fischaugenkamera basierende Verfahren auf einen zweiten Rechner verlagert. Im Framework sind somit zwei Komponenten verblieben, die nun für die Kommunikation mit der verteilten Komponente verantwortlich sind.

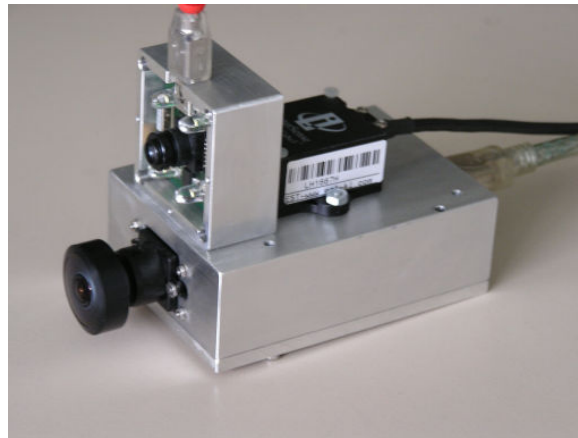


Abbildung 43: Versuchsaufbau mit zwei Kameras und Inertial-Sensor

Dabei wurden die Verfahren, die auf der perspektivischen Kamera arbeiten, wie das Framework am IGD entwickelt, während die anderen Komponenten teilweise von externen Partnern entwickelt wurden. Diese verteilte Entwicklung wird besonders durch die offene Struktur des Frameworks unterstützt.

Während das Multithreading der Applikation auf einem Desktopprozessor mit Hyperthreading zu hohen Frameraten der einzelnen Komponenten und der Applikation verhelfen, ist der Vorteil auf einem Mobilprozessor ohne zusätzliche virtuelle oder reale Prozessoren nicht spürbar. Vielmehr führt der Synchronisationsoverhead erst einmal zu Performanzeinbußen. Zusätzliches Tuning des Tracking-Frameworks ist notwendig, wobei eine gute Balance zwischen der Framerate einer Komponente und des Gesamtsystems gefunden werden muss.

4.4 Präsentation

Die Präsentation virtueller Informationen in Augmented Reality kann, je nach Anwendung, unter mehreren unterschiedlichen Gesichtspunkten betrachtet werden. Während bei vielen Anwendungen die Darstellung von 3D Geometrien, die mit der realen Umgebung registriert sind, im Vordergrund steht, dient in anderen Anwendungen AR der Interaktion mit Informationsobjekten, die ebenfalls mit der Umgebung des Anwenders registriert sind. Im zweiten Fall spielt die Darstellung von 3D Geometrien nur eine untergeordnete Rolle. Vielfach kann hier eine zweidimensionale Darstellung helfen, Inhalte besser zu erkennen.

Die einfache Darstellung der virtuellen Objekte über dem realen Hintergrund ermöglicht es dem Anwender nicht, die räumliche Relation zwischen realen und virtuellen Objekten einzuschätzen. Hier fehlen sämtliche Tiefenhinweise, die der Anwender aus der realen Welt kennt, so zum Beispiel Verdeckungen zwischen den Objekten oder der Schattenwurf eines Objekts auf ein anderes.

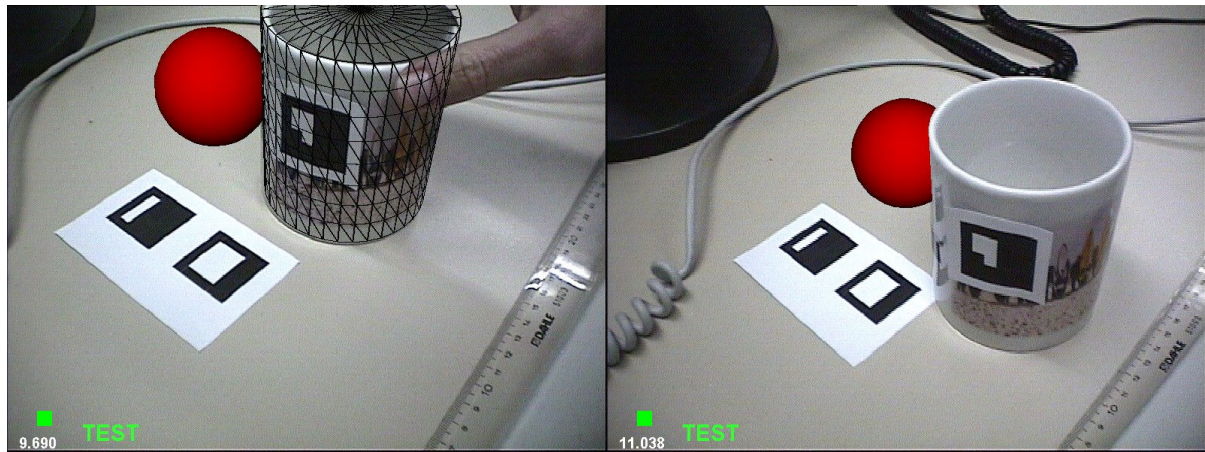


Abbildung 44: Verdeckungsdarstellung über Phantomgeometrie

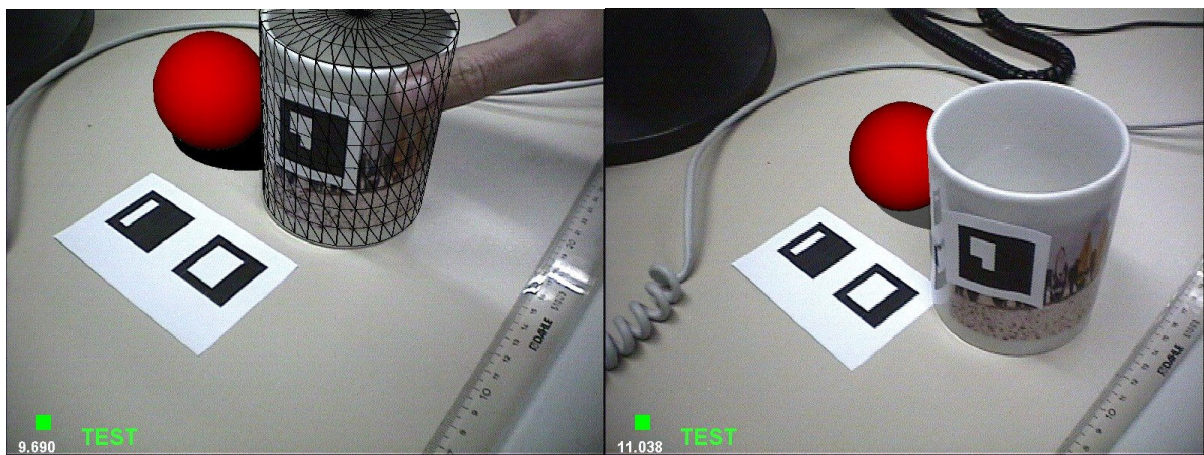


Abbildung 45: Verdeckungen und Volumenschatten

4.4.1 Darstellungsmodi

In der Charakterisierung von Azuma [Azuma97] ist eines der Kriterien für Augmented Reality die Registrierung der Informationen mit der Umgebung in 3D. Informationsobjekte werden in fester räumlicher Relation zur realen Umgebung dargestellt. Die Erkennbarkeit und Interaktion mit der virtuellen Umgebung wird hierdurch allerdings oftmals erschwert. Wird beispielsweise ein Video oder Text präsentiert, ist der Inhalt in der perspektivischen Sicht schlechter zu erkennen als in einer orthographischen 2D Sicht. Hinzu kommt, dass sich der Betrachter relativ zum Informationsobjekt bewegt, wodurch zusätzliche Darstellungsfehler bedingt durch die meist geringe Größe des Videos im Bild und die relativ geringe Auflösung des Displays entstehen. Obwohl das Video mit einem Ort der realen Umgebung in 3D verknüpft ist, kann es also sinnvoll sein, die räumliche Verknüpfung beim Abspielen des Videos zugunsten der besseren Erkennbarkeit aufzuheben. Das gleiche gilt für Bedienelemente, wie Menü oder Slider. Das Selektieren eines Bedienelementes, das eine feste Position im 2D Bildraum besitzt, ist wesentlich einfacher, als die Durchführung der Selektion in der 3D Umgebung, zumal sich der Benutzer auch noch relativ zu dieser Umgebung bewegt.

Die drei Darstellungsmodi, die Billinghamurst [BILL98] unterscheidet (siehe auch Abschnitt 2.3.2), können hier helfen, sinnvolle Darstellungsmethoden für die unterschiedlichen Informationsobjekte zu definieren.

Bezogen auf das obige Beispiel der Präsentation eines Videos bietet sich in diesem Fall die display-fixe Darstellung an. Der Anwender hat jederzeit den bestmöglichen Blickwinkel auf das Video. Für die Zeit der Präsentation ersetzt das Video weitestgehend die reale Umgebung. Werden dagegen dreidimensionale Geometrien zur Erklärung eines Arbeitsablaufs eingeblendet, müssen diese mit der realen Umgebung registriert werden. Diese Annotationen können nur dann sinnvoll interpretiert werden, wenn der räumliche Bezug bestehen bleibt. Ein rotierender Pfeil, der das Lösen einer Schraube andeutet, ist nur dann eindeutig, wenn auch erkennbar ist, auf welche Schraube sich die Anweisung bezieht.

Ein Ziel bei der Realisierung der Präsentationskomponente war die Wiederverwendbarkeit der Präsentationsobjekte für alle unterstützten Modi. Während head- und body-stabilized Darstellungen eher den Charakter von 2D-Darstellung besitzen, ist die Präsentation in der world-stabilized Methode schon durch die Registrierung mit der dreidimensionalen realen Umgebung oft mit der Darstellung von 3D Geometrien verbunden. Zur Erfüllung dieser Anforderung gibt es nun hauptsächlich zwei alternative Ansätze.

Die erste Möglichkeit ist die Generierung und Darstellung der Präsentationsobjekte durch zwei unterschiedliche funktionale Einheiten, einen 2D und einen 3D Renderer. Beide erzeugen aber die Darstellung aus der gleichen Beschreibung. Dies ist der Ansatz, wie er beispielsweise in DWARF realisiert wurde.

Die zweite Möglichkeit ist, die komplette Darstellung an eine 3D-Engine zu delegieren. Diese kann mit einem geringen Overhead auch die Objekte der 2D Darstellung erzeugen. Dieser Ansatz wurde auch bei der Realisierung der Benutzeroberfläche von Mac OS X durchgezogen. Hier ist die komplette Darstellung, also auch der 2D Desktop, OpenGL basiert. Der Leistungszuwachs bei 3D Grafikprozessoren war in den letzten Jahren um ein vielfaches größer als der Zugewinn bei Hauptprozessoren. Dies gilt nicht nur für den Desktopbereich sondern setzt sich auch bei Notebooks und Tablet PCs fort. Schwächere mobile Rechner können mittlerweile mit 3D Hardwarebeschleunigung aufwarten und sogar für Handys und PDAs sind Prozessoren für die Beschleunigung von 3D Funktionen angekündigt. Unterstützt wird dieser Trend durch die OPENGL-ES Initiative, mit dem Ziel, eine vereinfachte Version der standardisierten Grafikschnittstelle auf Geräten der Handy und PDA Klasse zur Verfügung zu stellen.

Ein zusätzliches Argument für die durchgängige Verwendung der 3D Darstellung ist, dass die 2D Funktionalitäten z.B. in OpenGL oftmals weniger gut unterstützt werden, als entsprechende 3D Techniken. Dadurch kann es sogar zu Performanzeinbußen kommen.

In der Präsentationskomponente werden alle darzustellenden Objekte als Knoten des Szenengraphs generiert. Kernstück dieser Komponente ist die VRML-Engine *Avalon* [BEHR04]. Über die API der Engine können zur Laufzeit beliebige Knoten erzeugt und gelöscht, ihre Felder gesetzt und Routen zwischen den Feldern erzeugt werden. Zusätzlich können Standardgrafikformate wie VRML97 und X3D gelesen werden. Das Rendering in Avalon geschieht auf Basis des Open-Source Szenengraphen OpenSG [REIN04]. Die Szene in OpenSG besteht grundsätzlich aus einem Vordergrund, dem Sze-

nengraphen und einem Hintergrund. Wird ein nichtdurchsichtiges Display verwendet, wird die reale Umgebung als Video-Hintergrund dargestellt.

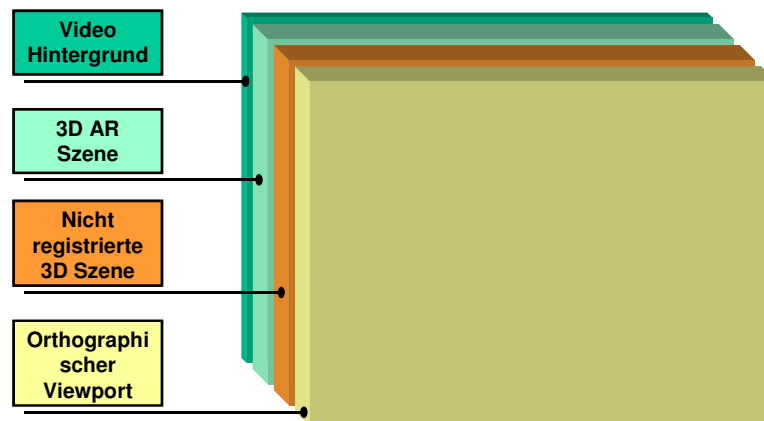


Abbildung 46: Standardmäßige Konfiguration der Darstellungslayer

Die Kombination von 2D- und 3D-Inhalten erfolgt über die Verwendung mehrerer Viewports, die in Schichten übereinander liegen. Jeder Viewport erhält dabei einen anderen Teil des Szenengraphen zugeteilt, der dann dediziert für die Darstellung einer Schicht zuständig ist.

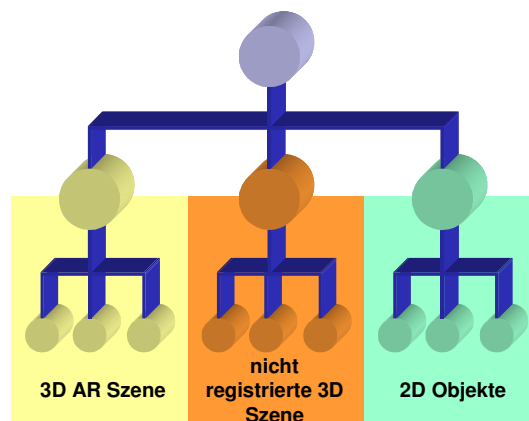


Abbildung 47: Szenengraphstruktur

Jeder Viewport besitzt eine eigene virtuelle Kamera. Während die 2D Sicht durch eine orthographische Projektion und die nicht registrierte 3D Sicht mittels einer frei parametrierbaren, perspektivischen Kamera realisiert sind, wird die virtuelle Kamera in der registrierten Sicht je nach verwendeter Displaytechnologie unterschiedlich gewählt und parametrierbar. Beim Einsatz eines nicht durchsichtigen Displays, also einer Video-See-Through Brille oder eines TabletPCs, werden die Parameter von der verwendeten Kamera vorgegeben. In diesem Fall ist ein einfaches perspektivisches Kameramodell ausreichend. Maßgeblich sind allein die intrinsischen Parameter der realen Kamera.

Im Falle der Verwendung eines See-Through-Displays muss das komplette System bestehend aus Display, Tracking-System und dem optischen System des Benutzers in der virtuellen Kamera abgebildet werden. Abbildung 46 und Abbildung 47 zeigen beispielhaft die Standardkonfiguration der Präsentationskomponente, es sind allerdings beliebige weitere Konfigurationen mit zusätzlichen Layern

möglich, die von einer zusätzlichen Komponente über die in Abschnitt 4.2.3 beschriebenen Mechanismen erzeugt und angesteuert werden können.

4.4.2 Visueller Realismus

Während der visuelle Realismus für viele Augmented Reality Anwendungen nicht von übergeordneter Bedeutung ist, können bestimmte Aspekte der Darstellung zu einer Verbesserung bei der Szenenwahrnehmung und somit der Leistung des Benutzers führen. Zusätzliche depth cues können zudem die Interaktion mit dem System erleichtern.

Oft wird an die visuelle Darstellung in AR Systemen die Anforderung physikalisch plausiblen Verhaltens formuliert. Zwei Aspekte sind in diesem Zusammenhang besonders hervorzuheben, die korrekte Behandlung von Verdeckungen zwischen realen und virtuellen Objekten und die Beleuchtung der virtuellen Szene inklusive der Darstellung von Schatten.

Ein Grundsatz bei der Realisierung jedes Verfahrens zur Steigerung des visuellen Realismus muss sein, dass solche Funktionalitäten im Falle der Nichtnutzung keine oder im schlimmsten Falle nur sehr geringe Rechen- und Speicherressourcen verbrauchen.

Für die korrekte Darstellung von Verdeckungen zwischen realen und virtuellen Objekten gibt es grundsätzlich zwei unterschiedliche Ansätze, den bildbasierten und den modellbasierten Ansatz. Beim bildbasierten Ansatz werden aus Kamerabildern pixelweise Tiefeninformationen gewonnen, beispielsweise durch die Gewinnung von Tiefeninformationen aus Stereobildpaaren [BREEN96] oder die Verwendung spezieller Sensorik, wie der Z-Cam [IDDAN01]. Neben einem nicht unerheblichen zusätzlichen Hardwareaufwand, der den Einsatz solcher Verfahren im mobilen Umfeld ausschließt, arbeiten diese Verfahren meist nur im Nahbereich befriedigend. Der modellbasierte Ansatz erfordert dagegen, dass alle realen Objekte, die mit der virtuellen Szene interagieren können auch als virtuelles Modell vorliegen. Diese Modelle können allerdings relativ grob ausfallen und es ist auch nicht notwendig, dass die realen Materialeigenschaften nachgebildet werden. Beispiele für die Verwendung von Verdeckungsobjekten finden sich beispielsweise in [REIN98].

Bei beiden Verfahren wird vor dem Rendern der sichtbaren Szene der Tiefenpuffer der Grafik-Hardware mit den Tiefenwerten der realen Szene gefüllt. Im bildbasierten Ansatz mit dem Kamerabild, im modellbasierten Ansatz dagegen wird die virtuelle Nachbildung der realen Umgebung in den Tiefenpuffer gerendert.

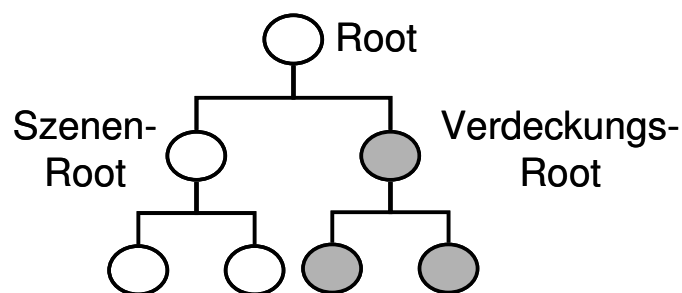


Abbildung 48: Struktur des Szenengraphen mit Verdeckungsdarstellung

Beim modellbasierten Verfahren ist es notwendig, dass vor der Darstellung der sichtbaren Szene alle Verdeckungsgeometrien in den Tiefenspeicher gerendert werden. In der Präsentationskomponente wird dies durch einen zusätzlichen Render-Pass sichergestellt. Die Unterteilung des Szenengraphen in eine sichtbare Szene und die Verdeckungsobjekte, siehe Abbildung 48, ermöglicht das wechselseitige Aktivieren der Teilbäume im jeweils richtigen Render-Pass. Die Deaktivierung des Root-Knotens eines Teilbaums verhindert, dass dessen Kinderknoten traversiert und in diesem Fall gerendert werden (Abbildung 49).

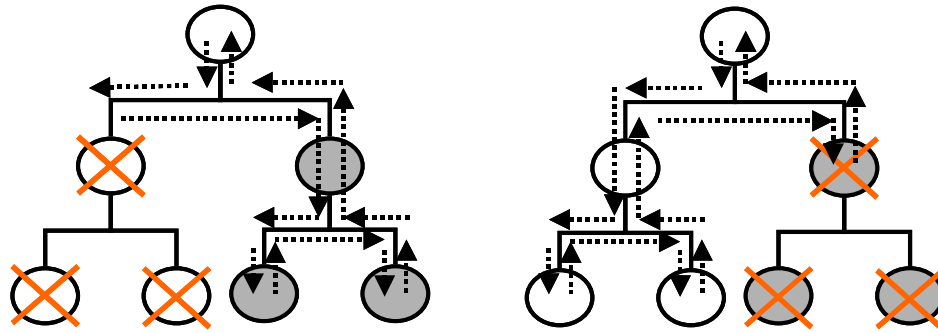


Abbildung 49: Traversierung der Teilbäume für Verdeckungs- (links) und Szenendarstellung (rechts)

Der Aufwand für das Umschalten ist somit also minimal. Ist keine Verdeckungsszene vorhanden, besitzt der Root-Knoten des Teilbaumes keine Kinder und der erste Verdeckungspass beim Rendern ist deaktiviert. Es entsteht somit keinerlei zusätzlicher Rendraufwand.

4.4.3 Anbindung Device-Interface - Präsentation

Bei der Verbindung von Tracking- bzw. Interaktions-Geräten und der Präsentation in mobilen AR Anwendungen sind einige Besonderheiten zu beachten, die sich im Vergleich der Verwendung von solchen Geräten in eher stationären Installationen ergeben.

Eine Beobachtung beim Einsatz eines AR-Systems ist, dass die Präsentations-Daten und die Tracking-Konfiguration oftmals eine sehr unterschiedliche Gültigkeitsdauer besitzen. So sind beispielsweise bei der AR basierten Unterstützung einer Service-Aufgabe für jeden Schritt des Arbeitsablaufs neue Informationen zu präsentieren, die typischerweise disjunkt mit den Anweisungen zu den vorherigen Schritten sind. Dagegen kann die Tracking-Konfiguration so lange beibehalten werden, bis signifikante Unterschiede in der Umgebung vorgenommen wurden. Ist eine Abdeckung zu entfernen, so sind erst die Befestigungen zu lösen und anschließend die Abdeckung selbst, wodurch das bisher verborgene Innere der Anlage sichtbar wird. Während also die präsentierten Informationen mit jedem der Schritte wechseln, wird die Tracking-Konfiguration erst mit dem Entfernen der Abdeckung geändert. Schon dieses einfache Beispiel zeigt, dass eine Trennung von Präsentations-Daten und der Tracking-Konfiguration notwendig ist.

Der zweite Punkt betrifft die Verwendung der Tracking-Informationen. Die in Abschnitt 4.3.1 präsentierten Geräteschnittstellen, wie IDEAL oder OpenTracker stellen dem System einzelne Funktionalitäten eines Eingabegerätes jeweils als abstraktes Gerät zur Verfügung. In einem rein szenengraphbasierten System werden die daraus resultierenden abstrakten Geräte direkt mit Feldern von Knoten verbun-

den. Ein Gerät, das Rotations- und Translations-Daten liefert, kann so einen Transformationsknoten ansteuern und die darunter hängenden Geometrieknoten transformieren (Abbildung 50).

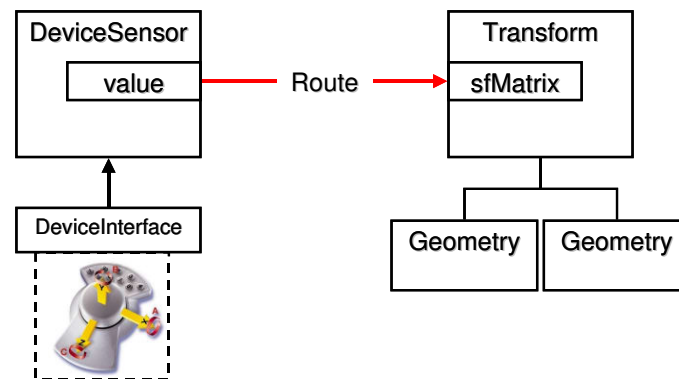


Abbildung 50: Geräteanbindung auf Szenengraphenebene

Jedes abstrakte Gerät wird auf Szenengraphenebene durch einen Knoten repräsentiert, der die Geräteschnittstelle ausliest. Nach außen werden die Werte des Gerätes über entsprechende Felder zur Verfügung gestellt. Über Routen werden die Werte dann zu den Zielknoten propagiert.

Allerdings können Tracker-Transformationen oder Triggerwerte nur an Knoten propagiert werden, die ebenfalls im Graphen hängen. Soll die Applikationslogik selbst angesteuert werden, ist diese ebenfalls über Knoten des Graphen zu realisieren. Hierzu stehen vor allem Scriptknoten zur Verfügung, in denen der Applikationscode realisiert ist. Bei der Änderung eines Input-Feldes wird der damit assoziierte Scriptcode ausgeführt.

Da in der vorliegenden Architektur quasi jede Komponente Zugriff auf die Tracking-Daten benötigt, ist die Anbindung des Tracking-Frameworks auf dieser hohen Systemschicht realisiert. Jede Komponente ist dadurch selbst verantwortlich, wie sie die anliegenden Daten verwendet.

Die Präsentationskomponente interpretiert dabei die Daten direkt als Transformationen im Szenengraph, wobei unterschieden wird zwischen der Transformation des ersten logischen Sensors, die als Position und Orientierung der Kamera relativ zur Umgebung interpretiert wird und den weiteren Transformationen, die dynamisch mit beliebigen Knoten der Szene verbunden werden können. Somit können neben der Bewegung des Benutzers durch die Szene auch Bewegungen von Objekten relativ zu diesem Referenzkoordinatensystem verfolgt werden. Ein Beispiel hierfür ist das Entfernen einer Abdeckhaube bei der Durchführung einer Wartungsarbeit. An der Abdeckung angebrachte virtuelle Objekte sollten sich in diesem Fall mit der Abdeckung bewegen. Beispielsweise sollte sich ein Verdeckungsmodell der Haube synchron mit der realen Haube bewegen so, dass diese weiterhin Teile der virtuellen Szene verdeckt.

Grundsätzlich ist dieses Problem mit der Verwendung eines Szenengraphen bereits gelöst. Die entsprechenden Szenenteile, die bewegt werden sollen, werden in einen Teilszenengraphen zusammengefasst, der unter einem Transformationsknoten hängt. Wird vom Tracking-System erkannt, dass sich das betroffene Objekt bewegt, wird die Transformation des entsprechenden virtuellen Objekts nachgeführt, reale und virtuelle Welt bleiben eine Einheit.

Allerdings müssen Schatten- und Verdeckungsobjekte ebenfalls unter die beschriebenen Transformationsknoten gehängt werden. Konsequenz ist, dass Schatten- und Verdeckungs-Objekte an beliebigen Stellen des Szenengraphen auftreten können. Es muss somit während der Traversierung des Baumes für jeden Knoten entschieden werden, ob er dargestellt werden muss oder nicht. Da auch bei dieser Realisierung die Szene mehrfach gerendert, also der Baum auch mehrmals traversiert werden muss, entsteht durch jeden zusätzlichen Knoten in einem Pass zusätzlicher Aufwand.

Die hier vorgestellte Realisierung beruht auf der Aufteilung des Szenengraphen in spezielle Teilbäume, wie bereits in Abschnitt 4.4.1 beschrieben. Allerdings werden dort keine dynamischen Szenen berücksichtigt. Die Verbindung von Tracking-Sensoren und Szenengraphknoten wird über dedizierte Knoten vorgenommen. Unterhalb des Root-Knotens wird für jeden verwendeten Sensor ein eigener Subgraph angelegt (Abbildung 51).

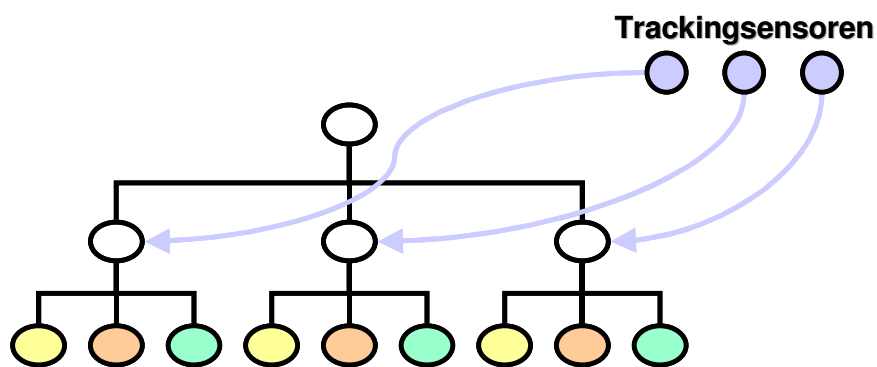


Abbildung 51: Verbindung von Teilszenengraphen und Tracking-Sensoren

Jeder Subgraph wiederum beinhaltet Teilbäume für die sichtbare Renderszene, die Schattengeometrien und zusätzlich einen eigenen Graphen für die Selektion. Je nach Bedarf ist eine feste Menge an Knoten, und damit der darunter hängenden Teilbäume, zu aktivieren oder zu deaktivieren.

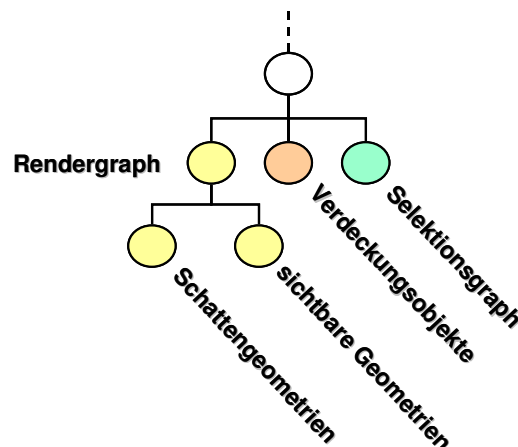


Abbildung 52: Struktur der Teilbäume

A priori ist bereits bekannt, welche Knoten umgeschaltet werden müssen, das ganze kann also dementsprechend effizient gehandhabt werden. So werden Teilbäume auch nur bei Bedarf aufgebaut und überflüssige Tests können vermieden werden.

4.5 Einbettung von AR Inhalten in Applikationen

Grundsätzlich können zwei verschiedene Typen der Integration von AR Inhalten in eine Applikation unterschieden werden. Damit verbunden sind auch zwei grundverschiedene Sichtweisen, wie AR eingesetzt wird. Die Applikation kann einerseits vollständig in AR integriert sein oder aber AR Inhalte stellen nur einen Teil der Gesamtapplikation dar. Der AR-Viewer ist also entweder selbst die Applikation oder integriert sich in eine Applikation.

4.5.1 AR-Viewer als Applikation

Der erste Ansatz, die vollständige Integration in ein AR System, impliziert, dass der AR-Viewer alle benötigten Dokumenttypen anzeigen können muss (Abbildung 53). Das gilt für verschiedene Formate von Textdateien ebenso, wie für Multimedia-Objekte und Zeichnungen.

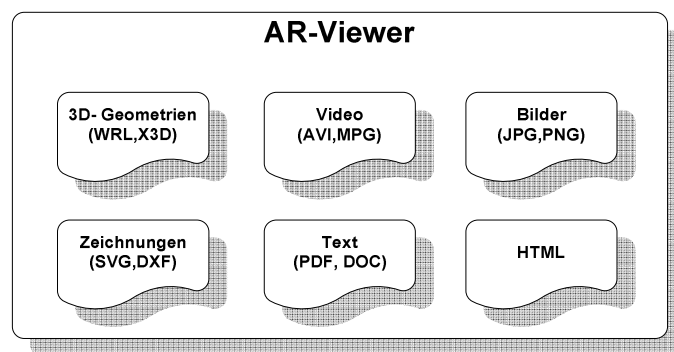


Abbildung 53: AR-Viewer stellt alle unterstützen Dokumenttypen dar

Die Unterstützung der verschiedenen Dokumenttypen erfolgt dabei entweder in der Applikation selbst oder in externen Programmen. Die Unterstützung verschiedener Dokumenttypen innerhalb der Applikation ist dabei mit erheblichem Mehraufwand verbunden und beinhaltet meist nur den gerade aktuellen Stand des Dokumenten-Typs. Ein Versionswechsel beim unterstützten Dokumententyp erfordert somit auch Änderungen am Programm.

Im Gegensatz zu vielen Standard-Bildformaten und einigen Multimedia-Typen, deren Spezifikation offen ist und für die oftmals bereits Bibliotheken oder Softwareentwicklungswerkzeuge existieren, sind viele der Formate gar nicht oder nur unzureichend dokumentiert. Abgesehen davon ist die direkte Unterstützung von umfangreichen Typen, wie Multimedia-Formaten, extrem aufwändig.

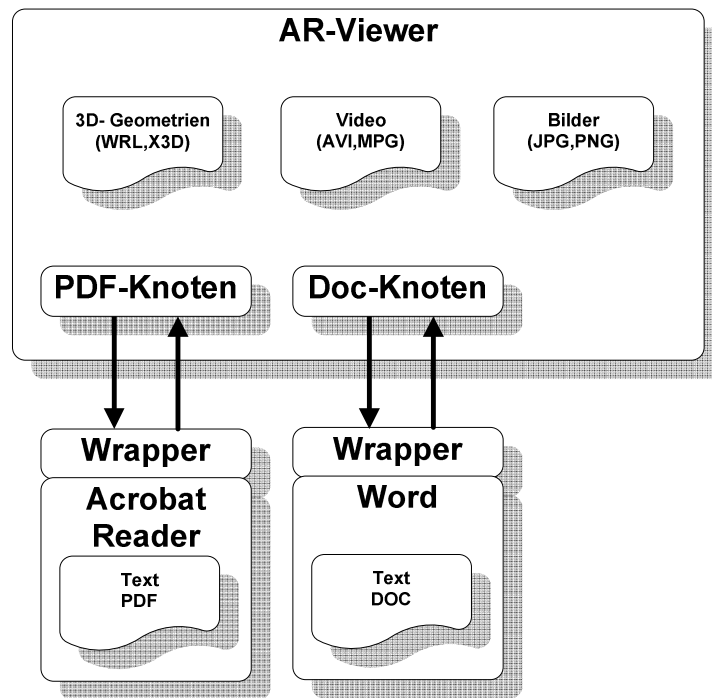


Abbildung 54: Integration von Dokumenttypen über Wrapper und entsprechende Szenengraph-Knoten

Alternativ bietet es sich deshalb an, die Verarbeitung der Dokumente im jeweiligen, dafür zuständigen Programm zu belassen und nur dessen Darstellung abzugreifen und diese „fernzusteuern“ (Abbildung 54). Ein Wrapper bettet hierfür ein ActiveX-Control ein und stellt dessen Funktionalität dem AR-Viewer zur Verfügung. Dieser Ansatz wird in [GEI03] verwendet, um beispielsweise Flash-Animationen in einer 3D-Gui darzustellen. Ein Flash-Player als ActiveX-Control liefert die Animation frameweise als Pixelbild, dass der AR Viewer als Textur auf ein entsprechendes Polygon aufbringt. Die Steuerung des ActiveX-Controls kann über ebenfalls in der Textur enthaltene Bedienelemente erfolgen. Die Position eines 3D-Mauszeigers auf dem Polygon wird dabei in die Pixel-Koordinaten des ActiveX-Controls umgerechnet und entsprechende Maus-Events erzeugt. Alternativ lassen sich auch zusätzliche Bedienelemente hinzufügen, die bereits im Wrapper verarbeitet werden, der dann Steuerkommandos für das ActiveX-Control generiert.

Allerdings erfordert auch die Einbettung von Viewern über Wrapper, dass für jeden Dokumenttypen ein neuer Wrapper geschrieben werden muss. Schon der Wechsel auf eine neuere Version eines Viewer-ActiveX kann zur Notwendigkeit von Anpassungen führen.

Eine weitere mögliche Alternative der Darstellung verschiedener Dokumenttypen bietet das Öffnen einer selbstständigen Viewer-Applikation über entsprechende System-Aufrufe. Der Aufwand der Realisierung besteht hier hauptsächlich im Bereitstellen einer Liste der Dokumenttypen und der damit verknüpften Programme und Aufrufparameter. Als problematisch bei dieser Art der Realisierung erweist sich der vollständige Verlust der Kontrolle über den Ablauf des Programms.

4.5.2 Einbettung von AR in eine Applikation – AR-Browser

Geht man davon aus, dass eine AR Präsentation nur eine zusätzliche Darstellungsvariante für Informationen ist, führt dies dazu, AR Inhalte als einen zusätzlichen Dokumententyp zu betrachten. Der AR-

Viewer ist in einer solchen Architektur nicht mehr die Applikation, sondern stellt einer umgebenden Applikation die Präsentation von AR Inhalten zur Verfügung.

Daraus lässt sich ein Paradigma ableiten, nach dem eine AR Präsentation von Daten eine zusätzliche, alternative, Sicht auf diese Daten darstellt. Informationen, die ohne AR als Text, Bild, Video oder 3D-Geometrie ohne räumlichen Bezug dargestellt werden können, werden durch AR in den realen räumlichen Bezug gesetzt. Im Sinne einer Document-View-Controller Architektur wird ein AR-View auf die Daten bereitgestellt.

Da zu einer AR-Präsentation mehrere unterschiedliche Daten notwendig sind, wie 3D-Geometrien und Tracking-Informationen, fasst ein AR Dokument genau genommen mehrere Dokumente zusammen. Durch die Interaktivität des Mediums AR handelt es sich dabei um ein dynamisches Dokument.

Die Applikationsschnittstelle AR-Browser erlaubt die Integration von AR Inhalten in Webbasierte Applikationen. Die Inhalte einer solchen webbasierten Anwendung werden in Form dynamischer HTML-Seiten auf dem Server erzeugt. Zusätzlich können aber auch andere Dokumenttypen in diese Seiten eingefügt oder von ihnen verlinkt werden.

Sieht man Augmented Reality als zusätzlichen Service innerhalb einer solchen Anwendung wird eine Schnittstelle benötigt, die es erlaubt von den HTML-Inhalten in eine AR-Szene und wieder zurück zu wechseln. Genau diese Anforderung kann mit der AR-Browser Schnittstelle erfüllt werden.

Die im BMBF-Projekt ARVIKA entstandene System-Architektur ist darauf ausgelegt, einen mobilen Arbeiter mit kontextabhängiger Information zu versorgen. Das komplette System ist aufgeteilt in eine Client Server Struktur, wobei der Client als Viewer für die verschiedenen Inhalte dient, die vom Server bereitgestellt und von diesem aufbereitet werden. So wird die Präsentation von Inhalten bereits auf dem Server an die jeweilige clientseitige Hardwarekonfiguration angepasst. Je nachdem, ob ein Head-Mounted-Display oder ein Tablett PC verwendet wird, werden unterschiedliche Benutzerschnittstellen erzeugt.

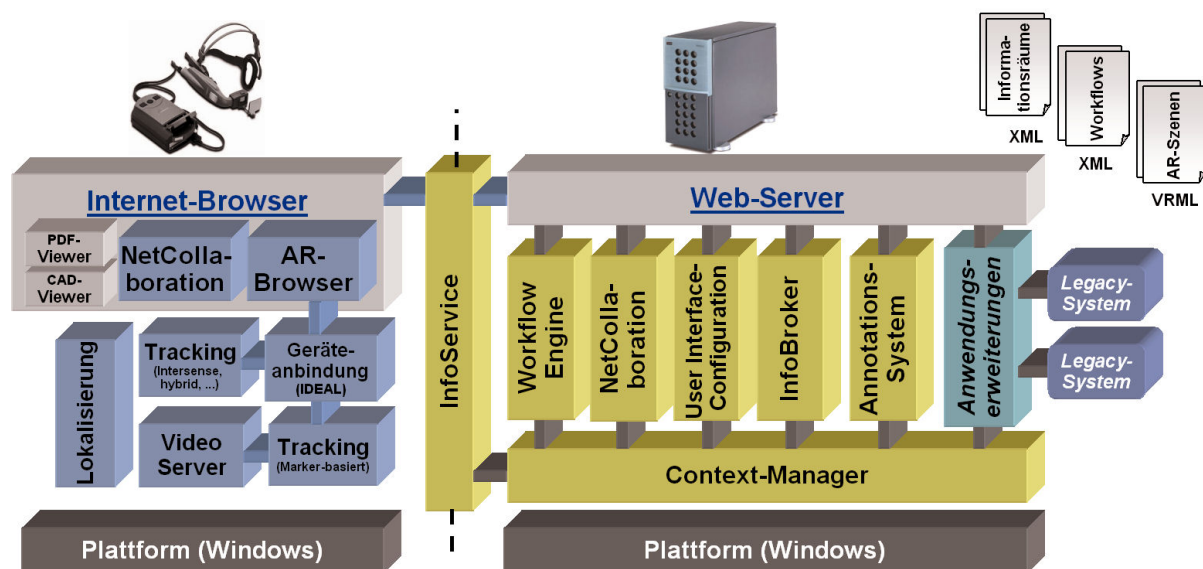


Abbildung 55: ARVIKA-Architektur zur Unterstützung von mobilen Systemen

Die eigentliche AR Basisfunktionalität verbirgt sich in der Komponente AR-Browser. Diese ist als ActiveX-Komponente ausgelegt, so dass sie sich einfach in webbasierte Applikationen einbetten lässt. Konzeptionell stellt in dieser Architektur Augmented Reality einen zusätzlichen Dokumententyp dar. Ähnlich, wie es Viewer-Plugins für verschiedene Dokumententypen, wie PDF oder DXF gibt, ist der AR-Browser ein Viewer für AR-Dokumente (Abbildung 55). Augmented Reality wird somit als Ergänzung zur „klassischen“, eher textbasierten Dokumentation betrachtet.

Die Softwarearchitektur des Gesamtsystems soll hier unter dem speziellen Gesichtspunkt der eigentlichen Augmented Reality Kernfunktionalität betrachtet werden. Diese ist zusammengefasst in der clientseitigen Komponente AR-Browser.

In der Gesamtsystemarchitektur fügt sich der AR-Browser in eine webbasierte Gesamtarchitektur ein, d.h. die Inhalte werden über das HTTP-Protokoll von einem Server bereitgestellt. Als Client dient ein Internet-Browser, der die dynamisch generierten HTML-Seiten anzeigt. Zusätzlich besitzt der Internet-Browser Plugins für die Anzeige verschiedener Dateiformate. Wird die Präsentation von Informationen in einer AR-Umgebung gewünscht, wird der AR-Browser instanziiert. Dies geschieht über das Referenzieren des ActiveX-Controls in einer HTML-Seite über das <OBJECT>-Tag und die entsprechende, eindeutige ID des Controls.

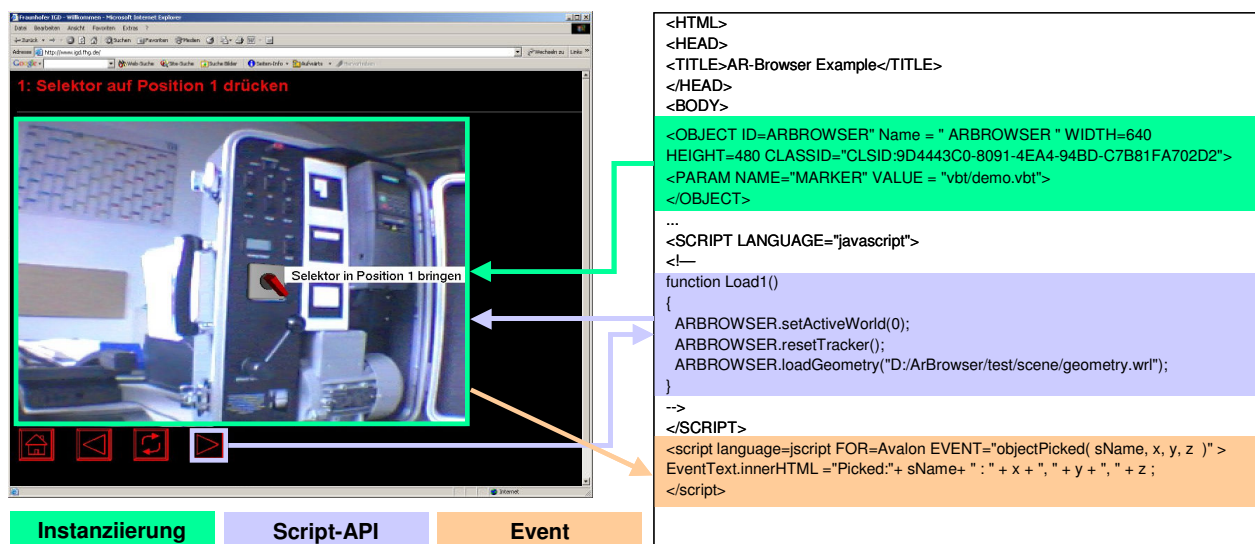


Abbildung 56: AR-Browser Integration in HTML-Inhalte

Ist das AR-Browser Objekt instanziiert, kann es über ebenfalls in HTML eingebettete Scriptaufrufe gesteuert werden (Abbildung 56). Der AR-Browser besitzt hierfür eine Script-Programmierschnittstelle, die weit reichenden Zugriff auf alle internen Funktionalitäten erlaubt. Neben vielen festen Zugriffsmethoden können auch sämtliche im System vorhandenen Attribute der einzelnen Komponenten gelesen und gegebenenfalls auch geschrieben werden. Dieses Vorgehen verlangt zwar viel Wissen über die inneren Strukturen des Gesamtsystems, es erlaubt allerdings auch weitestgehende Freiheit bei der Programmierung des Systems zur Laufzeit.

Mit der Realisierung als ActiveX-Control geht allerdings die Abhängigkeit zur Plattform Microsoft Windows und Internet-Explorer einher. Im Gegenzug kann so allerdings das Einbetten in fast beliebige Windows Anwendungen sehr einfach vorgenommen werden, die ebenfalls als Container für solche

Controls dienen, so dass diese Anwendungen auf einfache Weise um AR Funktionalitäten erweitert werden können.

Intern stellt die AR-Browser ActiveX-Schnittstelle nur eine dünne Schicht zwischen Anwendung und der eigentlichen AR-Kernfunktionalität dar, so dass der Kern auch auf anderen Systemplattformen, wie Irix und Linux verwendet werden kann. Der AR-Core wird hierfür in den entsprechenden Applikationsrahmen, beispielsweise eine Qt-Applikation eingebunden. Der Kern selbst enthält die Funktionalitäten für Tracking, Präsentation, Interaktion und die Anbindung für Tracking- und Interaktionsgeräte sowie den Zugriff auf unterschiedliche Videoquellen. Alle systemnahen Funktionalitäten, wie eben das Videograbbing sind in entsprechenden Abstraktionen verborgen, die eine Portierung auf andere Zielplattformen ohne Modifikation des eigentlichen Kern-Codes erlauben. Das szenengraphbasierte Renderingsystem OpenSG und die darüber liegende VRML/X3D-Engine Avalon sind ebenfalls bereits für die Verwendung auf unterschiedlichen Zielplattformen, wie Windows, Linux und IRIX vorgesehen.

4.6 Zusammenfassung

Die Konzeption eines Frameworks und der zugrunde liegenden Architektur sind komplexe Problemstellungen mit weit reichenden Auswirkungen auf die Arbeit mit solchen Systemen. Aufgrund der in Kapitel 3 beschriebenen Erhebung und Analyse der Anforderungen an ein solches Framework für die Realisierung von mobilen AR Anwendungen, können Designentscheidungen oder Entscheidungen für oder gegen bestimmte Vorgehensweisen und Detaillösungen relativ fundiert gefällt werden.

Der Aufbau der Gesamtarchitektur und der darin enthaltenen Komponenten, sowie deren grundlegende Eigenschaften folgen dabei vielfach sehr pragmatischen Lösungsansätzen. Dabei wurden allerdings stets die großen Ziele Wiederverwendbarkeit von Komponenten und beliebige Erweiterbarkeit des Systems im Auge behalten.

Zusätzlich zu diesen Basis-Diensten und –Funktionalitäten werden auch neuartige Komponenten, wie das erweiterte Weltmodell eingeführt, dass eine effiziente Erstellung von Informationsräumen erlaubt. Zusätzlich bilden die atomaren Einheiten dieser Informationsräume, die Informations-Objekte auch die Grundlage für weitergehende Dienste, wie die Informationsfilterung.

Relativ breiten Raum bei der Betrachtung von Architekturen für AR Systeme nehmen typischerweise die Schnittstellen zum Tracking und der Aufbau der verwendeten Tracking-Subsysteme selbst ein. In der vorgestellten Architektur stehen hierfür mehrere Alternativen zur Verfügung, die sowohl die Anbindung bestehender Systeme, wie OpenTracker oder IDEAL, erlauben als auch eine eigenständige Lösung zur komponentenbasierten Realisierung von hybriden Tracking-Verfahren. Bei diesen zusammengesetzten Tracking-Systemen liegt der Schwerpunkt auf der Kombination unterschiedlicher visionbasierter Verfahren mit dem Ziel, einzelne Verfahren so einzusetzen, dass ihre Schwächen weitestgehend eliminiert werden und ihre Stärken besonders zur Geltung kommen.

Auch die Komponente zur visuellen Präsentation erhält in einem AR System, bei dem der Schwerpunkt auf der visuellen Synthese von realer und virtueller Welt liegt, besondere Bedeutung. Neben den Standardtechniken, wie der Verwendung eines szenengraphbasierten Renderingsystems, werden wei-

tergehende Techniken zur Erhöhung des visuellen Realismus, hier im Speziellen die Darstellung von Echtzeitschatten zwischen realen und virtuellen Objekten betrachtet. Außerdem wird auf die Einbettung von nicht dreidimensionalen Daten in die Visualisierung eingegangen, da solche Multimedia-Informationen, wie Videos, Standbilder, Zeichnungen aber auch Texte in den hier betrachteten Applikationsdomänen große Bedeutung zukommt.

Abschließend wird auf die Frage eingegangen, wie AR Inhalte in andere Applikationen einzubetten sind. Entweder als eigenständiger Viewer, der dann selbst auch die Darstellung der unterschiedlichsten Dokumententypen übernehmen muss oder als Control, dass in andere Anwendungen eingeklinkt werden kann, der Weg, der mit dem AR-Browser beschriftet wurde.

5 Interaktionskonzept für mobile AR Anwendungen

In einem Mixed Reality Szenario bewegt sich der Benutzer durch die reale Umgebung, die mit virtuellen Informationen angereichert ist. Die virtuellen Objekte sind dabei mit der realen Umgebung registriert, sie scheinen mit dieser eine Einheit zu bilden. Zusammen bilden die virtuellen Szenenanteile eine dreidimensionale Benutzerschnittstelle durch die sich der Benutzer bewegen kann.

Durch die Registrierung zwischen virtuellen Objekten und realer Umgebung verschmilzt die Benutzerschnittstelle mit der realen Umgebung: „The Interface is everywhere!“ [SCHMAL01]. Der Zugriff auf die Informationen, die mit realen Objekten und Orten verknüpft sind, erfolgt ortsbezogen.

Benötigt beispielsweise ein Servicetechniker Informationen zu einem bestimmten Bedienelement einer komplexen technischen Anlage, findet er die Informationen, die mit diesem Bedienelement verknüpft sind über den Ort des realen Bedienelements. Um diesen Zugriff auf Informationen effizient zu gestalten, werden vor allem im mobilen Umfeld neuartige Interaktionstechniken benötigt.

Für die Interaktion mit solchen räumlich verteilten Benutzerschnittstellen fehlen geeignete Interaktionsgeräte ebenso, wie die dazugehörigen Interaktionsmetaphern. Während für immersive VR Umgebungen wie CAVE oder Powerwall spezielle Eingabegeräte und darauf abgestimmte Interaktionsmetaphern entwickelt wurden, die eine weitgehend natürliche Interaktion mit der virtuellen Szene ermöglichen, existieren auf diesem Gebiet bezüglich mobiler AR Umgebungen nur vereinzelte Forschungsarbeiten.

Ein charakteristisches Merkmal der meisten Interaktionstechniken für immersive, virtuelle Umgebungen ist die Verwendung eines präzisen Tracking-Systems, das zur Verfolgung des Interaktionsgerätes notwendig ist. In immersiven VR Installationen stellt diese Abhängigkeit im Allgemeinen auch kein Problem dar, da vom Vorhandensein einer entsprechenden Tracking-Infrastruktur ausgegangen werden kann. Der Einsatz von elektromagnetischen, ultraschall-basierten oder stationären optischen Tracking-Verfahren ist unter den meist gegebenen kontrollierten Laborumgebungen ohne weiteres möglich.

Auch für die Realisierung von Augmented Reality ist das Vorhandensein einer funktionierenden Tracking-Infrastruktur Grundvoraussetzung. Soll das AR-System allerdings mobil eingesetzt werden, geht dies mit einer Beschränkung der einsetzbaren Tracking-Technologien einher. Mit den heute verfügbaren Tracking-Technologien, die auch in mobilen Anwendungen eingesetzt werden können, ist eine direkte Umsetzung der meisten Interaktionstechniken aus dem VR Bereich nicht ohne weiteres möglich. Da aber auf der Anforderungsseite die Interaktion mit einer dreidimensionalen Benutzerschnittstelle steht, die ebenso wie in immersiven VR Anwendungen den Anwender umgibt, gilt es adäquate Interaktions-Techniken bereitzustellen, die ähnlich effizientes Arbeiten wie in diesen VR Anwendungen erlauben.

5.1 Grundlagen der Interaktion in virtuellen Umgebungen

Bei der Beschreibung der Interaktion in einem Softwaresystem kann zwischen den Interaktionsaufgaben, mit denen ein Benutzer konfrontiert wird und den Interaktionstechniken, die das System zur Bewältigung dieser Aufgaben bereitstellt, differenziert werden. Grundsätzlich kann zwischen den folgenden fünf Basis-Interaktionsaufgaben unterschieden werden, die zusammengesetzt auch zur Beschreibung komplexer Interaktionsaufgaben verwendet werden können [FOL90]:

- Navigieren
- Positionieren / Orientieren
- Selektieren bzw. Deselektieren
- Quantifizieren
- Texteingabe

Bei der Gestaltung der Interaktionstechniken sind mehrere grundlegende Anforderungen, wie die Aufgabenangemessenheit, die schnelle Erlernbarkeit oder die Reiz-Reaktions-Korrespondenz zu berücksichtigen.

5.2 Interaktionsaufgaben in mobilen AR Anwendungen

Abhängig von einem bestimmten Aufgabengebiet und der verwendeten Technologie, besitzen die fünf Basis-Interaktionsaufgaben unterschiedliches Gewicht. So ist beispielsweise das Navigieren durch die 3D-Szene, also die Positionierung und Orientierung der Kamera, in rein virtuellen Umgebungen eine sehr wichtige Aufgabe. In AR dagegen besitzt diese Aufgabe so gut wie keine Bedeutung, da die virtuelle Kamera fest mit der Blickrichtung des Benutzers verbunden ist. Wird diese Verbindung aufgelöst, geht die Registrierung zwischen realer und virtueller Welt verloren und man kann genau genommen nicht mehr von AR sprechen.

Das Positionieren und Orientieren von Objekten, Polygonen oder einzelnen Eckpunkten, also die Manipulation von Objekten, ist vor allem für Modellierungsaufgaben von Bedeutung. Einige Beispiele, wie Tinmith-Metro [PIEK01b] oder Amire [ZAUN03], deren Schwerpunkt auf der Erstellung von 3D Szenen in einem mobilen AR System liegt, zeigen dass die Manipulation von Objekt-Geometrien auch in AR Anwendungen von großer Bedeutung sein kann. Für die hier betrachteten Anwendungsgebiete besitzt die Manipulation der virtuellen Szene allerdings relativ wenig Relevanz.

Die Interaktionsaufgabe der Quantifizierung beschreibt das Setzen eines Wertes innerhalb eines vorgegebenen Wertebereichs. Meist wird diese Eingabe verwendet, um Objektattribute wie die Transparenz zu ändern oder einen Parameter zur Steuerung des Systems, beispielsweise die maximale Sichtweite des Anwenders zu ändern. Besonders der letztere Aspekt, die interaktive Steuerung von System-

funktionen spielt in AR oft eine große Rolle, wenn der Benutzer Anpassungen des Systems an seine Umgebung vornehmen muss.

Die Eingabe von Text bezieht sich auf Inhalte, die nicht vom System interpretiert werden. Ein Beispiel sind hier Annotationen, die ein Anwender an einem Objekt anbringt und mit einem textuellen Kommentar versieht. Das Fehlen einer Tastatur erfordert in immersiven VR und mobilen AR Anwendungen, dass hier alternative Methoden, wie eine virtuelle Tastatur, angeboten werden müssen, die aber den Komfort einer realen Desktoptastatur nicht ersetzen können. Aus diesem Grund wird das Anbringen textueller Annotationen vielfach durch andere Eingabemodalitäten, wie die Sprachaufzeichnung, ersetzt.

Die Selektion virtueller Objekte dient als Grundlage aller weiteren Interaktionsaufgaben, die ein aktives Objekt für die Durchführung benötigen. Generell kann zwischen zwei Formen der Selektion unterschieden werden, der Objektselektion und der Menüselektion. Während bei der Menüselektion ein Element aus einer eng begrenzten, festen Menge ausgewählt wird, kann der Umfang dieser Menge bei der Objektselektion sehr groß werden und ihr Inhalt sich dynamisch ändern.

Die Selektion von Objekten wird in mobilen AR Anwendungen, wie sie hier betrachtet werden, intensiv verwendet. Bevor die einzelnen Dokumente, die mit einem Informationsobjekt verlinkt sind, für den Anwender sichtbar werden, muss das Informationsobjekt aktiviert werden. Da die Objekte in der dreidimensionalen Umgebung um den Anwender verteilt sind, werden hier ähnliche Techniken, wie in VR benötigt. Ist ein Informationsobjekt selektiert und beinhaltet mehrere Verweise auf Dokumente, folgt eine weitere Selektionsaufgabe. Während bei diesem zweistufigen Verfahren die erste Selektion über eine Objektselektion zu realisieren ist, kann die zweite Stufe als Menüselektion behandelt werden. Es wird also in einer zusammengesetzten Aktion Gebrauch von beiden Selektionsarten gemacht.

Während in Desktopanwendungen die Selektion auch in dreidimensionalen Anwendungen meist über eine Maus, selten eine Spacemouse, erfolgt, werden in VR und AR andere Techniken benötigt, die eine intuitivere Selektion erlauben. Die gebräuchlichsten Selektionstechniken in immersiven VR Umgebungen sind die Kollisionserkennung, die Strahlselektion oder auch Ray-Casting und die Image-Plane Selektion.

Die Kollisionserkennung ermöglicht dem Anwender die Selektion von Objekten innerhalb seiner Armreichweite. Ein getracktes Eingabegerät wird mit einem virtuellen Objekt verknüpft, das als Echo in der Darstellung dient und gleichzeitig über die Kollision mit anderen Objekten der Szene deren Selektion auslöst. Eine typische Verwendung dieser Technik ist das Greifen virtueller Objekte über einen getrackten Datenhandschuh, bei der das virtuelle Pendant der Hand auch als Kollisionsobjekt dient. Nachteil dieser Techniken ist, wie bereits eingangs erwähnt die Beschränkung des Arbeitsbereichs auf die tatsächliche Reichweite des Anwenders. Ray-Casting und Image Plane Interaction sind zwei Techniken, die diese Einschränkung überwinden.

Beim Ray-Casting, oder auch Strahlselektion, wird ein Selektionsstrahl in die virtuelle Szene geschickt [HINCK94]. Die Richtung des Strahls wird dabei durch ein in sechs Freiheitsgraden getracktes Interaktionsgerät, wie den Polhemus Stylus [POLH] gesteuert. Das Ergebnis einer solchen Selektion sind ein Schnittpunkt zwischen Selektionsstrahl und der getroffenen Objektgeometrie, sowie der Bezeichner des getroffenen Objekts.

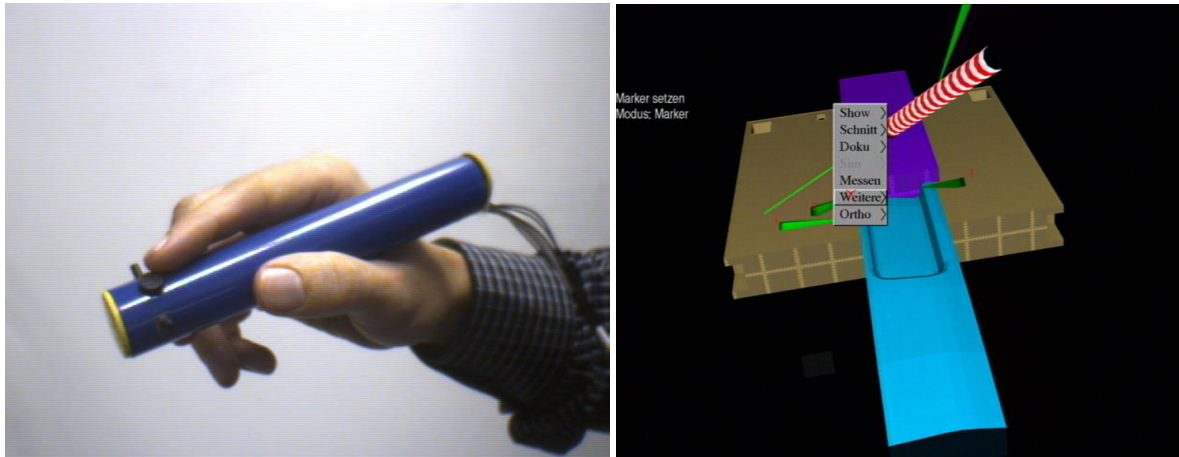


Abbildung 57: Flystick 6DOF Interaktionsgerät und dessen Echo in VR Anwendung

Ein Problem dieses Verfahrens stellt die Selektion sehr kleiner oder weit entfernter Objekte dar, da bereits kleine Bewegungen des Interaktionsgeräts in zunehmender Entfernung große Winkelbewegungen verursachen. Einen Lösungsansatz für diese Problematik bietet die Cone Selection [LIAN94]. Hierbei wird der Selektionsstrahl durch einen Konus ersetzt, der mit zunehmender Entfernung zum Anwender im Durchmesser zunimmt. Als Kandidaten für eine Selektion kommen alle Objekte in Frage, die sich zumindest teilweise innerhalb des Volumens befinden. Da dies mehrere Objekte sein können, ist zusätzlich eine Auflösung der daraus resultierenden Mehrdeutigkeit durchzuführen.

Bei der Image Plane Interaktion interagiert der Benutzer mit der 2D Projektion der Szene in der Bildebene [PIERCE97]. Über verschiedene Gesten, die mit der Hand im Blickfeld gebildet werden und den damit verbundenen Metaphern können Punkte oder Regionen im Bildraum definiert werden, die das zu selektierende Objekt beinhalten.

Voraussetzung für die Realisierung der Image Plane Interaktion ist, dass der Kopf und die Hände des Anwenders getrackt werden. Allerdings muss nur die Blickrichtung in 6D verfolgt werden, bei den Händen selbst ist dann eine zweidimensionale Lokalisierung in der Bildebene ausreichend. Insgesamt ergibt sich hieraus ein zweistufiger Prozess bei dem zuerst durch das Tracking des Kopfes und der damit verbundenen Erfassung der Blickrichtung der Referenzrahmen für die darauf folgende Interaktion in der Bildebene gebildet wird.

5.3 Interaktionstechniken in mobilen AR Anwendungen

Die Interaktion in mobilen AR Anwendungen unterscheidet sich von der Interaktion in immersiven virtuellen Welten vor allem durch die verfügbaren Interaktions- und Tracking-Geräte. Während der Großteil der immersiven VR Installationen auf Groß- und begehbaren Mehrseitenprojektionen, so genannten CAVEs basieren, ist in AR die Verwendung von Head-Mounted-Displays notwendig, um dem Benutzer Informationen direkt ins Blickfeld einblenden zu können.

Im Gegensatz zur Vielzahl verfügbarer Tracking-Verfahren, die in stationären VR Installationen verwendet werden können, sind in mobilen AR Anwendungen einzig visionbasierte Tracking-Verfahren in der Lage, die Anforderungen an Genauigkeit und Anzahl an Freiheitsgraden zu erfüllen. Grundsätz-

lich ermöglichen auch diese Verfahren, dass neben Position und Blickrichtung des Benutzers auch Interaktionsgeräte mit der gleichen Kamera verfolgt werden. Allerdings ist der Arbeitsbereich eines solchen Interaktionsgeräts auf das Blickfeld der Tracking-Kamera beschränkt. Befindet sich das Interaktionsgerät außerhalb dieses Blick-Volumens ist eine Bestimmung von Position und Orientierung nicht mehr möglich. Neben dieser Begrenzung des Arbeitsvolumens besteht ein weiteres Problem in der relativ geringen Bildauflösung, mit der dieses Volumen abgetastet wird. Objekte die von einem Tracking-Algorithmus sicher erfasst werden sollen, müssen auch eine gewisse Größe im Bildraum besitzen.

Angenommen, die Tracking-Kamera besitzt VGA Auflösung, also 640x480 Pixel, so deckt ein Pixel des Kamerabilds bei einem horizontalen Blickwinkel von 40° einen Winkel von 0,0625° ab. Umgekehrt entsprechen 16 Pixel einem Winkel von 1°. Ein Objekt mit einem Durchmesser von 2 cm beansprucht in der Entfernung von 1m, was ungefähr Armeslänge entspricht, einen Bildausschnitt von 1,15° und wird von der Kamera mit ca. 18 Pixeln erfasst. Da vor allem die Winkelfehler bei derart kleinen zu trackenden Objekten dramatisch zunehmen, ist die Steuerung eines Selektionsstrahls mit solchen Techniken nur unter Verwendung größerer Marker möglich. Allerdings bedeutet die Verwendung entsprechend größerer Marker auch eine Reduzierung des verbleibenden Sichtfelds für die sichtbare reale Umgebung.

Generell ist die Bestückung der Hände des Anwenders mit künstlichen Markern kritisch zu betrachten, da sich die Handhabung vor allem bei größeren Markern ungünstiger gestaltet und diese auch bei der Durchführung von Aufgaben in der realen Umgebung erheblich behindern.

Neben der Position der Marker im Bild ist bei einigen Interaktionstechniken, zum Beispiel bei der Selektion im 3D Raum, auch die Bestimmung einer Orientierung von Bedeutung. Diese kann entweder aus der Orientierung von einzelnen Markern abgeleitet werden oder aus der relativen Position mehrerer Marker zueinander gebildet werden. Die Bestimmung der Orientierung eines Markers ist vor allem bei spitzen Winkeln zwischen Marker-Ebene und Blickvektor schwierig und wenig robust. Ebenso problematisch ist die Bestimmung eines Richtungsvektors aus der Position zweier Marker, wenn deren Abstand sehr gering ist. Einzig das Bilden eines Richtungsvektors zwischen Augenposition des Anwenders und einem Marker im Bildbereich ermöglicht eine recht präzise und robuste Grundlage für die Selektion im 3D Raum.

Aus ergonomischer Sicht stellt sich vor allem die notwendige Armhaltung des Benutzers, der zu trackende Objekte in sein Blickfeld bringen muss, als problematisch heraus. Der Arm muss annähernd horizontal nach vorne gestreckt werden, wobei neben den kleineren Muskelpartien des Unterarms und der Hand auch die großen Muskelpartien des Oberarmes und im Schulterbereich benötigt werden, was zu einer recht unnatürlichen Arbeitshaltung und damit verbunden zur schnellen Ermüdung der beteiligten Muskelpartien führt (siehe Abbildung 58 rechts).

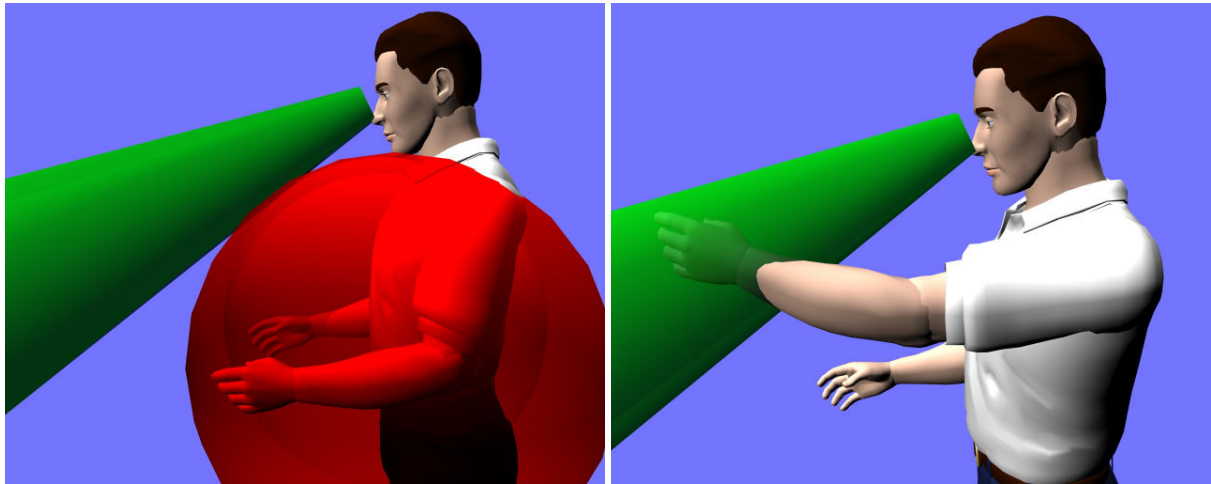


Abbildung 58: Komfortable Arbeitsbereiche (rot) und Blickfeld (grün)

Abhilfe kann hierbei eine zweite Kamera bringen, die den körpernahen Bereich vor dem Benutzer im Blick behält. Ein Interaktionsgerät kann somit auch außerhalb des Blickfelds der ersten Kamera verfolgt werden. Da beide Kameras in fester Relation zueinander montiert sind, kann nach einer einmaligen, initialen Kalibrierung eine einfache Transformation zwischen den beiden Tracking-Systemen berechnet werden. Nachteil dieser Lösung ist aber, dass zusätzliche Hardware benötigt wird und der doppelte Aufwand bei der rechenintensiven Bildverarbeitung entsteht, so dass sich auch hieraus keine wirkliche Lösungsalternative für den mobilen Einsatz ergibt.

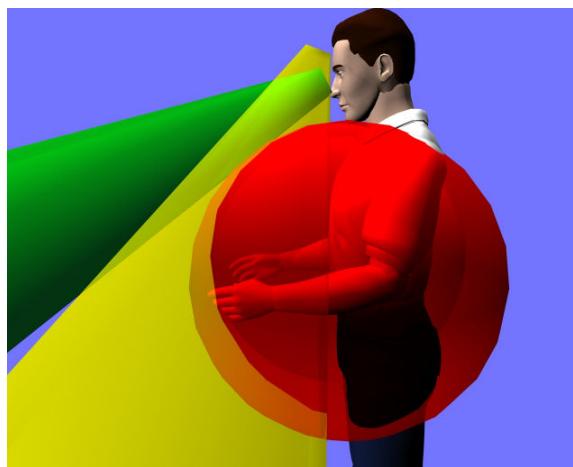


Abbildung 59: Visionbasiertes Tracking des Arbeitsbereichs über zusätzliche Kamera (gelb).

Image Plane Interaktionstechniken in Verbindung mit einem HMD erweisen sich als besonders schwer zu kontrollieren. Der Benutzer steuert über seine Kopfbewegungen das Referenzsystem, also den Bildausschnitt, in dem er die Interaktionen durchführen kann. Selbst kleine Kopfbewegungen führen dazu, dass sich Selektionsbereich und Umgebung relativ zu einander bewegen. Die Image Plane Interaktion über ein HMD in AR erfordert somit sehr viel Aufmerksamkeit und Konzentration des Anwenders für die Koordination von Kopf- und Handbewegungen. Aufmerksamkeit, die eigentlich für die reale Aufgabe benötigt wird und somit zu einer zusätzlichen mentalen Belastung führt.

Eine weitere Möglichkeit, die Selektion zu realisieren ist die so genannte Gaze-Selection, bei der ein Selektions-Strahl oder –Volumen entlang der Blickrichtung aus einem Punkt oder Bereich der Bild-

ebene gebildet wird. Bei der Verwendung eines getrackten HMDs kann der Anwender somit über Kopfbewegungen die Selektion von Objekten in seiner Umgebung steuern. Objekte, die sich im Selektionsbereich befinden, werden dabei als aktuell selektierbares Objekt vorgewählt, wobei das Auslösen der eigentlichen Selektion entweder über einen Tastendruck oder einen Sprachbefehl erfolgen kann. Alternativ kann die Selektion auch zeitgesteuert ausgelöst werden, wobei das längere Verharren des Selektionsbereichs über einem Objekt dessen Selektion auslöst, was allerdings auch zur ungewollten Selektion eines Objektes führen kann, wenn der Benutzer unbewusst darauf fokussiert.

Gaze Selektion kann also sehr einfach mit den vorhandenen Bordmitteln eines AR-Systems realisiert werden, erfordert aber wie die Image Plane Selektion im HMD viel Aufmerksamkeit des Anwenders. Unwillkürliche Bewegungen des Benutzers führen dabei leicht zum Verlust einer Selektion. Außerdem ist die Gaze-Selektion problematisch bei Objekten, die sehr viel Platz im Bildraum beanspruchen und somit die Selektion anderer kleinerer Objekte erschweren können. Neben der mentalen Belastung durch das gezielte Steuern der Blickrichtung führt die Gaze-Selektion auch zu einer starken Belastung der Muskulatur im Hals- und besonders im Nackenbereich und führt somit leicht zu Beschwerden in diesen Bereichen. Ungeachtet dieser Problematik bietet die Gaze-Selektion aber eine fast optimale Stimulus-Reiz-Korrespondenz, da keinerlei mentale Transformation zwischen der Bewegung des Anwenders und der daraus generierten Aktion des Rechners notwendig ist.

Einen speziellen Fall der Image Plane Interaktion stellt die Selektion auf einem PDA oder TablettPC mit dem Stift mit der Maus oder auf einem Notebook über ein Touchpad dar. Auch hier wird der Ausgangspunkt des Selektionsstrahls in der Bildebene definiert. Der Benutzer kontrolliert dabei mit der Hand, die den Rechner hält, die Blickrichtung der daran angebrachten Kamera. Mit der anderen Hand kann er im Blickfeld befindliche Objekte mit einem geeigneten 2D Interaktionsgerät anfahren und die Selektion auslösen. Auf einem TablettPC kann diese Aufgabe komplett mit der Stiftmetapher erfolgen. Ansonsten muss ein üblicherweise auf Notebooks verfügbares Touchpad oder ein anderer Mausersatz verwendet werden. Diese Eingabegeräte haben allerdings den Nachteil, dass die Positionierung relativ erfolgt, das Objekt also unter Umständen über einen langen Weg angefahren werden muss und auch der direkte Bezug zwischen Eingabe und Reaktion des Systems verloren geht. Ein TablettPC mit seiner absoluten Cursor-Positionierung direkt im Bild ist bei dieser Aufgabe weit überlegen.

Bei der Bedienung eines TablettPCs über die Positionierung eines Stifts auf dem Display handelt es sich um eine Variante der beidhändigen Interaktion, bei der typischerweise beide Hände unterschiedliche Aktionen durchführen. Die nichtdominante Hand führt den Rechner und gibt damit den Referenzrahmen für den Stift vor, der mit der dominanten Hand geführt wird. Die Bewegung des Rechners in die gewünschte Richtung stellt den Referenzrahmen für die präzise Ansteuerung mittels Pen bereit. Die Selektion wird somit in zwei aufeinander aufbauende Phasen zerlegt, eine gröbere Vorpositionierung und eine feine Präzisionspositionierung.

Alternativ kann auch eine Selektion, ähnlich der Gaze-Selektion, über einen festgelegten Bereich des Bildes erfolgen. Auch die oben beschriebenen Trigger-Mechanismen aus der Gaze-Selektion können hierbei Anwendung finden. Im Gegensatz zur zweistufigen Selektion über die Ausrichtung des Gerätes und die eigentliche Selektion über einen Pen oder ein Touchpad wird hier die gesamte Selektion einstufig durchgeführt. Aus ergonomischer Sicht stellt vor allem das hohe Gewicht von TablettPCs

und Notebooks ein Problem dar, dass ein Arbeiten über einen längeren Zeitraum erschwert, da die Armmuskulatur neben der Positionierungsaufgabe vor allem auch mit der Haltearbeit gefordert ist.

Als Erweiterung der oben beschriebenen Verfahren können TablettPCs, PDAs und Grafik-Tabletts auch in Verbindung mit einem HMD eingesetzt werden. Man spricht dann von einem hybriden User-Interface, bei dem die Interaktion über die 2D-Eingabe auf dem jeweiligen Gerät mittels Stift-Metapher erfolgt (siehe Abbildung 60).



Abbildung 60: Hybride User-Interfaces realisiert über PDA und Grafiktablett

Bei der Realisierung von hybriden User-Interfaces entsteht ein teilweise beträchtlicher Zusatzaufwand an benötigter Hardware. Es wird, abgesehen von der Variante mit dem Grafik-Tablett, ein zweiter vollwertiger Rechner benötigt, der die Benutzereingaben entgegen nimmt und einen Teil der Darstellung übernimmt und der mit dem eigentlichen AR Rechner kommuniziert (siehe Abbildung 61).

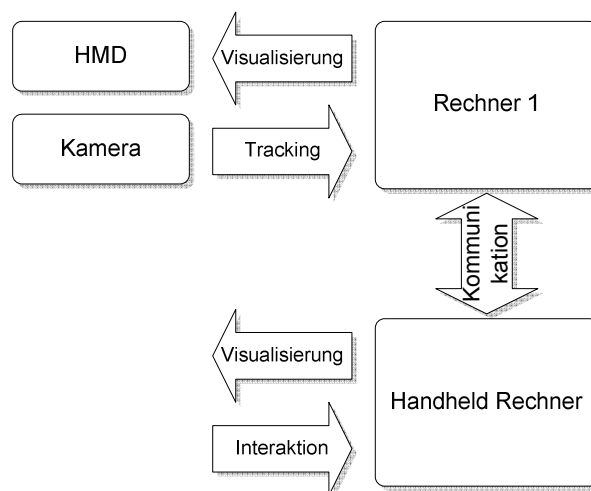


Abbildung 61: Schematischer Systemaufbau für die Realisierung eines Hybriden Userinterface mit einem zweiten Rechner

Zusätzlich entsteht für den Benutzer somit ein nicht unerhebliches Mehrgewicht, mit dem er hantieren muss. Da das Gerät nur zur Eingabe genutzt wird, bietet sich deshalb ein Handheldrechner an, der die gleiche Stift-Interaktionsmetapher bietet, aber erheblich kleiner und leichter ist (Abbildung 60 links). Ein weiterer Nachteil ist die schlechte Ablesbarkeit eines Rechnerdisplays durch ein HMD, in das bereits zusätzliche Informationen eingeblendet werden.

Alternativ lässt sich ein hybrides User-Interface auch mit einem Grafik-Tablett realisieren, das keine eigene Bild-Ausgabe bietet, dafür aber eine deutlich größere Eingabefläche gegenüber einem PDA besitzt (Abbildung 60 rechts). Dieser Ansatz entspricht dem Personal Interaction Panel des Studierstube Systems [SZAL97], was allerdings die Erfassung von Position und Orientierung des Tablett relativ zur Blickrichtung des Anwenders erfordert, damit im HMD eine entsprechende virtuelle GUI präsentiert werden kann, die auf das Tablett projiziert erscheint. Die Erfassung der räumlichen Relation zwischen HMD und Tablett kann über die Kamera erfolgen, die auch für das Head-Tracking verwendet wird. Hierbei ergeben sich allerdings wieder die gleichen Probleme wie bei der Image Plane Selektion, das Tablett muss sich dabei im Blickfeld der Kamera befinden und verringert somit den Bereich, der für die Sicht auf die reale Umgebung zur Verfügung steht. Auch die Arbeitshaltung wird somit beeinträchtigt, da entweder der Blick auf das Tablett gerichtet oder das Tablett in das Sichtfeld gebracht werden muss. Wird die virtuelle GUI in der Darstellung im HMD auf das Tablett projiziert, ist es allerdings aus Gründen der Sicht- und Erkennbarkeit der GUI-Elemente und somit auch der Bedienbarkeit unabdingbar, dass das Tablett eine exponierte Stellung im Blickfeld einnimmt.

Erfordert eine Anwendung, dass der Benutzer beide Hände für die reale Aufgabe zur Verfügung hat, bedeutet dies eine Beschränkung der Eingabemöglichkeiten über Sprache und Gaze-Selektion. Allerdings erfordern nur die wenigsten Tätigkeiten dauerhaft beide Hände, so dass zwischendurch auch wieder eine oder beide Hände zur Interaktion zur Verfügung stehen. Es ist dabei vor allem zu bedenken, dass die Verwendung herkömmlicher Dokumentation, sei es auf einem Rechner oder sogar in Papierform eine vollständige Unterbrechung der realen Tätigkeit erfordert. Alle bisher vorgestellten Interaktionstechniken stellen hier schon eine erhebliche Verbesserung dar, da der Benutzer nur in einem gewissen Maß von dieser Tätigkeit abgelenkt wird.

5.4 Universelles Interaktionsgerät für mobile AR Anwendungen

Aus den obigen Beschreibungen bestehender Interaktionstechniken für AR Anwendungen lässt sich ableiten, dass vor allem die Selektion in der dreidimensionalen Umgebung eine neue Klasse von Eingabegeräten erfordert. Verbunden mit darauf abgestimmten Interaktionstechniken kann die AR Unterstützung für weite Anwendungsfelder erheblich verbessert werden.

5.4.1 Anforderungen

Die Anforderungen an ein solches Interaktionsgerät ergeben sich einerseits aus den technischen Einschränkungen, die der Einsatz im mobilen Umfeld bedingt, also vor allem das Fehlen einer Tracking-Infrastruktur, die in einem größeren Arbeitsbereich Tracking-Informationen in sechs Freiheitsgraden bereitstellen kann. Zum anderen ergeben sich funktionale Anforderungen, die ein Interaktionsgerät

erfüllen soll, dass unter anderem der Selektion in der dreidimensionalen Umgebung des Anwenders dienen soll. Nicht zuletzt gilt es die ergonomischen Eigenschaften zu beachten, und hier vor allem die Belastung des Benutzers, sowohl physisch als auch mental.

Ziel der Entwicklung eines universellen Interaktionsgerätes ist es, mit einem einzigen Gerät alle Interaktionsaufgaben in einer mobilen AR Anwendung damit bewältigen zu können. Es sollten also keine weiteren Gerätschaften benötigt werden, um alle Systemfunktionen oder Funktionalitäten einer Anwendung zu bedienen. Dabei sollte das Eingabegerät allerdings nicht mit Funktionalitäten überladen werden, sondern vielmehr wenige optimierte Bedienelemente für unterschiedliche Aufgaben besitzen. Zusätzlich zur hardwareseitigen Betrachtung müssen auch die softwareseitigen Interaktionstechniken optimal auf dieses Gerät abgestimmt werden.

Da der Auslöser für die Entwicklung dieses Eingabegerätes das Fehlen geeigneter Geräte und Techniken für die 3D Selektion in mobilen AR Anwendungen war, ist dies auch die mit höchster Priorität verfolgte Anforderung an das Gerät. Unter Berücksichtigung der existierenden technischen und ergonomischen Randbedingung soll ein Anwender mit dem Gerät in die Lage versetzt werden, Objekte in seiner dreidimensionalen Umgebung mit diesem Gerät einfach zu selektieren.

Entscheidende Faktoren für die körperliche Belastung sind vor allem das Gewicht des Geräts und die Haltung in der es bedient wird. Hierbei ist vor allem zu betrachten, welche Muskelgruppen beteiligt sind. Basierend auf den Untersuchungen von Zhai [ZHAI96] sollte das Interaktionsgerät die Nutzung kleiner Muskelgruppen, wie der Finger unterstützen um optimale Performanz, vor allem bei den Selektionsaufgaben zu erzielen.

5.4.2 Konzept

Wie bereits aufgeführt, bringt die Selektion in 3D über Image Plane Techniken in einem HMD basierten mobilen AR System erhebliche Probleme mit sich. Vor allem die Bewegung des Referenzrahmens durch unwillkürliche Bewegungen des Kopfes und das unnatürliche Verharren des Kopfes in einer bestimmten Position zur Erlangung eines stabilen Referenzsystems sind hier die Hauptkritikpunkte. Es soll hier nun ein Weg aufgezeigt werden, wie auch in mobilen AR Anwendungen Ray-Casting Selektion unter Berücksichtigung der bestehenden ergonomischen und technischen Nebenbedingungen realisiert werden kann.

Die Selektion über Ray-Casting Techniken erfordert die Bestimmung des Ausgangspunktes des Selektionsstrahls und seiner Richtung. Es muss also eine 6D Pose bestimmt werden, die mit der Position und Orientierung des realen Interaktionsgeräts übereinstimmt. Woods realisiert die MagicMouse [Woods03], ein Interaktionsgerät, dass auf einem mit optischen Markern versehenen Handschuh basiert, der in einem fixen Volumen erfasst wird, dass von einer fest installierten Kamera vorgegeben wird. Piekarsky verwendet ebenfalls Marker an einem Handschuh [PIEK03] wobei hier unterschiedliche Marker an den Fingern mit verschiedenen Aktionen gekoppelt sind. Über die Marker können sowohl Position als auch Orientierung der Hand bestimmt werden. Hierfür ist allerdings ist die optimale Erkennbarkeit der optischen Marker im Kamerabild erforderlich. Um dies zu gewährleisten, muss der Anwender seine Gesten optimal im Blickvolumen der Kamera präsentieren, was bei der typischerweise am Kopf befestigten Kamera zu einer recht unnatürlichen und unbequemen Handhaltung führt. Ein

weiteres Problem stellt hierbei die große Fläche, die die Hände im Bildraum für optimale Erkennung der Gesten einnehmen müssen dar. Die Durchführung realer Arbeiten und die Wahrnehmung der realen Umgebung und der darin platzierten virtuellen Inhalte leidet hierdurch deutlich.

Ein gänzlich anderer Weg wird mit dem iOrb [Reit05] beschritten, einem Interaktionsgerät, das speziell für die Verwendung in mobilen AR Anwendungen konzipiert ist. Dieses Interaktionsgerät besteht hardwareseitig aus einem Rotationssensor und einem Taster (siehe Abbildung 62).

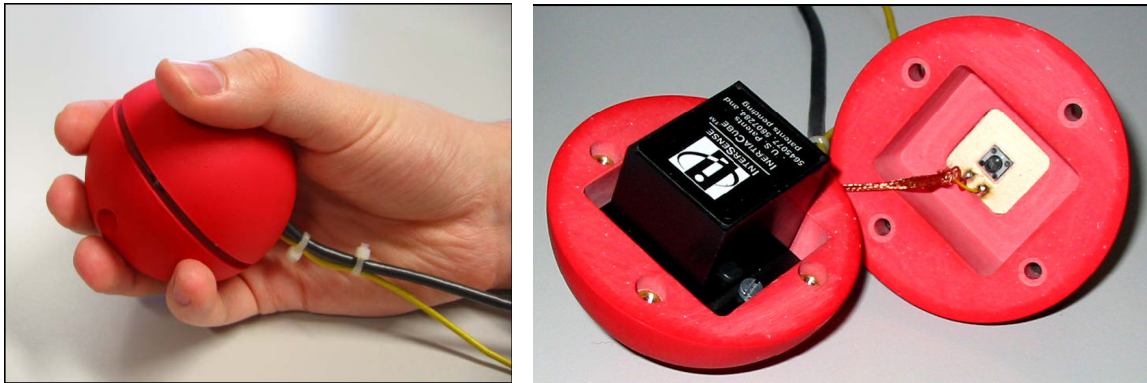


Abbildung 62: iOrb Interaktionsgerät [Reit05]

Ungeachtet des simplen Hardwareaufbaus ist hierbei vor allem die Ansteuerung des Raycastings zur Selektion durch die Rotationsmessungen interessant. Es werden dabei lediglich Rotationen um zwei Achsen betrachtet, die Pronation und Supination des Unterarmes und die Rotation um eine darauf senkrecht stehende Achse (siehe Abbildung 63).

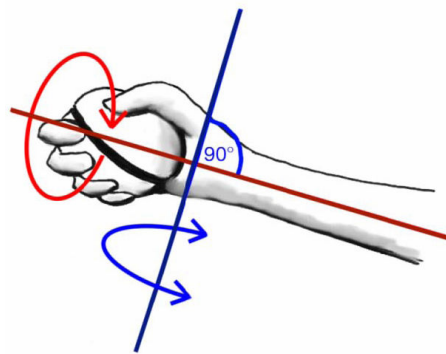


Abbildung 63: iOrb Freiheitsgrade

Das Ergebnis dieser Technik sind somit zwei unabhängige Messwerte, die sowohl zur Ansteuerung eines Selektionsstrahls als auch für die Interaktion mit Widgets verwendet werden können. Die Stärke dieses Verfahrens zeigt sich darin, dass sowohl die Selektion per Raycasting als auch die Bedienung von Widgets einheitlich gestaltet werden kann und hierfür sehr natürliche Bewegungen verwendet werden. Der Preis hierfür ist allerdings ein Verlust der Reiz-Reaktions-Korrespondenz, da Bewegung und intuitiv erwartete Reaktion des Systems nur bedingt übereinstimmen. Die Steuerung des Raycastings über die beiden Rotationen erfordert die Kenntnis und das Erlernen der zugrunde liegenden Metapher.

Soll die Selektion in einem mobilen AR System realisiert werden, ist die hier vorgeschlagene Methode, die komplexe Gesamtaufgabe in mehrere weniger komplexe Teilaufgaben zu unterteilen und diese mit geeigneten Methoden zu lösen. In der gestellten Aufgabe bietet sich nach Betrachtung bestehender Technologien zur Lösung solcher Aufgaben in anderem Kontext, wie VR oder stationären AR Systemen, die Aufteilung in die Bestimmung der Position des Interaktionsgeräts in 3D und davon getrennt die Bestimmung der Orientierung ebenfalls in 3D. Dieses Vorgehen ist nicht zu verwechseln mit den Untersuchungen, Rotations- und Translationsaufgaben getrennt von der linken und rechten Hand durchführen zu lassen [ZHAI98]. Vielmehr betrifft die Aufteilung in eine translatorische und eine rotatorische Aufgabe die Verteilung auf unterschiedliche Tracking-Technologien oder auch andere Strategien und Heuristiken zur Bestimmung der jeweiligen Werte.

Betrachtet man das Problem der Rotationsbestimmung isoliert, so bietet sich hier die Lösung über einen 3D Rotationssensor an, wie es auch im oben beschriebenen iOrb zum Einsatz kommt. In dieser Geräteklasse existieren mehrere integrierte Lösungen, die sich in der Anwendung vor allem bezüglich Empfindlichkeit, Präzision und Baugröße unterscheiden. Die Tracking-Systeme bestehen aus einem einzelnen kleinen Block, der die Beschleunigungssensoren enthält und leicht in ein Interaktionsgerät integriert werden kann. Der benötigte Strom kann über die USB Schnittstelle bezogen werden, über die ein solches Gerät an den Rechner anzuschließen ist. Die Sensoren sind in der Lage mit hoher Frequenz Messungen der Orientierung im Raum durchzuführen, so dass eine Anfrage an einen solchen Sensor immer ein sehr aktuelles Ergebnis liefert, das mit minimaler Latenz zur Verfügung steht. Die Messwerte sind allerdings immer relativ zu einem bestimmten Initialwert und zusätzlich gekennzeichnet durch eine Anfälligkeit für sich akkumulierende Fehler.

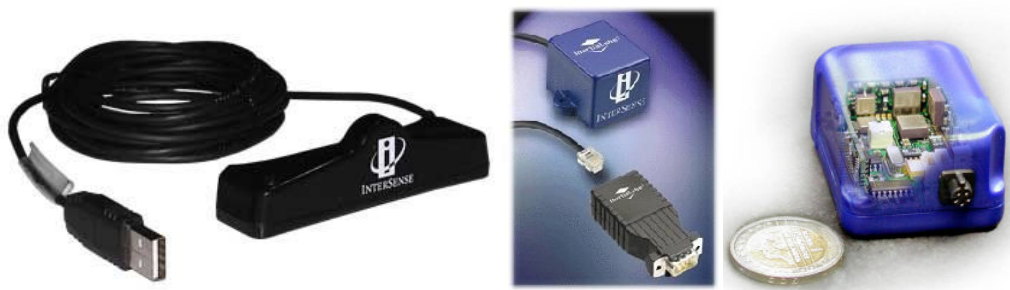


Abbildung 64: Verschiedene marktverfügbare Inertial-Trackingsysteme [Intersense, XSens]

Generell würde sich an dieser Stelle die Verwendung eines bereits komplett integrierten Interaktionsgerätes anbieten, allerdings erfüllen die verfügbaren Lösungen, die bereits einen Beschleunigungssensor besitzen, nicht die gestellte Anforderung nach drei Freiheitsgraden. Interaktionsgeräte wie die Gyration GyroRemote (siehe auch Abschnitt 2.3.3.2) sind vielmehr für Image Plane basierte Interaktionstechniken in 2D geeignet, da sie lediglich Koordinaten im 2D-Bildraum liefern können. Ein direkter Zugriff auf 3D Informationen oder die Rohdaten der Sensoren ist nicht möglich.

Auffällig ist allerdings die gute Reiz-Reaktions-Korrespondenz der GyroRemote, die über Drehungen der Hand die Steuerung des Cursors einer Desktopoberfläche erlaubt. Die Bildkoordinaten der GyroRemote werden als Ausgangspunkt der Strahlselektion in die virtuelle Szene verwendet. Geiger et al verwenden die GyroRemote für die Bedienung registrierter 3D Bedienoberflächen [GEIG03].

Image Plane Techniken, wie in dieser Anwendung erfordert vom Benutzer, wie bereits beschrieben die schwierige Koordination von Kopf- und Handbewegungen. Zur Überwindung dieses Nachteils ist „echte“ Strahlselektion notwendig, die unabhängig von der aktuellen Blickrichtung des Anwenders funktioniert.

Die Bestimmung der realen Position des Interaktionsgeräts erfordert ein zusätzliches Tracking-System, das den körpernahen Bereich abdeckt, in dem sich das Interaktionsgerät befindet. Aus bereits beschriebenen Gründen existiert allerdings kein System, das die gestellten Anforderungen erfüllt. Beobachtet man allerdings die Verwendung eines 6D Interaktionsgeräts bei Selektionsaufgaben in immersiven VR Anwendungen, so fällt auf, dass bei der Durchführung der Selektion das Gerät meist in annähernd der gleichen Position gehalten wird und der Selektionsstrahl überwiegend über Rotationen der Hand gesteuert wird. Der Benutzer nimmt hierfür meist eine für ihn angenehme Körperhaltung ein und steuert das Interaktionsgerät vor allen mit den kleinen Muskelgruppen der Hand und des Unterarms.

Aus dieser Beobachtung kann direkt die im Folgenden vorgestellte Realisierung von Selektion in mobilen AR Anwendungen abgeleitet werden, bei der auf ein Tracking der Position des Interaktionsgeräts verzichtet werden kann, wenn anhand eines Modells des Benutzers diese Position realistisch geschätzt oder vermessen werden kann. Während also die Orientierung des Interaktionsgeräts über Rotationsensoren gemessen wird, kann dessen Position als konstant für den Zeitraum der Selektion angenommen werden. Diese Modellierung beruht dabei auf der Annahme, dass ein Selektionsvorgang nur eine relativ kurze Zeitspanne in Anspruch nimmt und sich der Anwender bereits vorher dem Ort der Selektion zugewendet hat.

Die Position des Interaktionsgeräts kann relativ zur Kopfposition des Anwenders modelliert werden. Die Position entspricht somit dem Offset zwischen dem Tracking-System für die Bestimmung von Kopfposition und Blickrichtung und der angenommenen Position des Interaktionsgeräts. Während der Initialisierung der Selektion wird die zu diesem Zeitpunkt gemessene Position des Kopfes zur Initialisierung des Modells verwendet.

Für das dynamische Verhalten des Benutzermodells bestehen mehrere Alternativen, ähnlich der Unterscheidung bei den Darstellungsmodi zwischen display-fix und body-fix. Entscheidend hierfür ist, wie eine Rotation um die senkrechte Achse des Benutzers interpretiert wird. Diese kann entweder als Kopf- oder als Körperdrehung verarbeitet werden. Wird die Rotation als Körperdrehung interpretiert, wird das Benutzermodell ebenfalls mitrotiert, während beim Vorliegen einer Kopfdrehung das Benutzermodell in der Ausgangsposition verharrt.

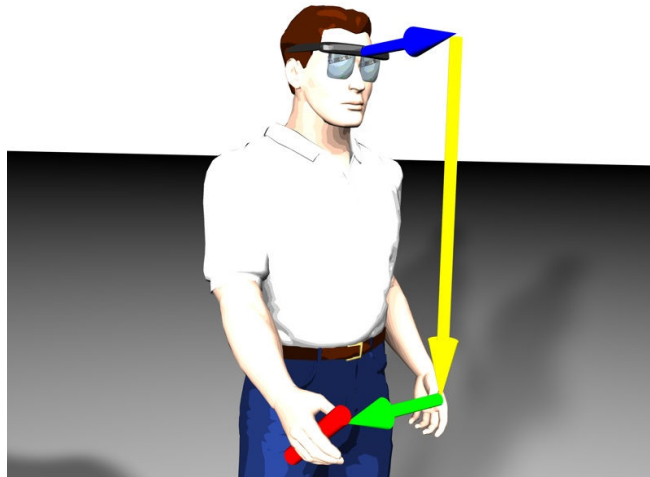


Abbildung 65: Transformation zwischen Tracking-System und Interaktionsgerät

Da das Benutzermodell mit relativen Werten zur aktuell gemessenen Kopfposition arbeitet ist es zu Beginn eines Selektionsvorgangs mit geeigneten Werten zu initialisieren. Erst durch diese Initialisierung erhält das virtuelle Interaktionsgerät eine Position und Orientierung in Weltkoordinaten. Diese Werte bilden die Ausgangslage für die Rotationsdaten der Beschleunigungssensoren.

Initialisierung:

M_K : Initiale Kopf-Pose des Benutzers

M_M : Offset zwischen Kopf des Benutzers und Interaktionsgerät

M_I : Initiale Pose des Interaktionsgeräts: $M_I = M_K * M_M$

Die Initialisierung der Pose des Interaktionsgeräts erfolgt zu einem Zeitpunkt t_0 . So dass M_I die Pose zum Zeitpunkt t_0 ist: $M(t_0)$.

Je nach dynamischem Verhalten des Benutzermodells ergibt sich daraus die Transformation des Interaktionsgeräts für den Zeitpunkt t_i wie folgt:

$M(t_i) = M(t_0) * R_S(t_i)$ für den Fall, dass das Körpermodell fix ist oder

$M(t_i) = M(t_0) * R_{KZ}(t_i) * R_S(t_i)$ für das dynamische Modell

mit $R_{KZ}(t_i)$ der gemessenen Rotation des Kopfes um die Körperachse zum Zeitpunkt t_i und $R_S(t_i)$ der gemessenen Rotation des Interaktionsgeräts zum Zeitpunkt t_i .

Da der Inertialsensor kontinuierlich die Rotationen des Interaktionsgerätes relativ zu einer arbiträren Anfangsorientierung misst, ist es zusätzlich notwendig die Richtung des Selektionsstrahls zu initialisieren, so dass dieser zu Beginn des Selektionsvorgangs eine definierte Richtung besitzt. Diese Initialisierung kann über die Blickrichtung des Benutzers erfolgen, so dass der Selektionsstrahl in seiner Anfangsrichtung immer im Blickfeld des Anwenders sichtbar ist.

Alternativ ist auch eine einmalige Initialisierung bei jeder Inbetriebnahme des AR Systems denkbar. Voraussetzung ist hier allerdings, dass der Benutzer sich kontinuierlich durch das gleiche Referenzkoordinatensystem bewegt, so dass diese Orientierung immer ihre Gültigkeit behält. Bewegt sich der Benutzer dagegen von einem Raum zum nächsten und wechselt dabei von einem relativen Koordina-

tensystem in ein anderes, verliert die initiale Orientierung mit hoher Wahrscheinlichkeit ihre Gültigkeit.

$R_S(t_i)$ in den obigen Gleichungen ist somit zu ersetzen durch

$$R_S(t_i)^{\wedge} = R_S(t_0)^{-1} * R_S(t_i)$$

so dass alle Messungen als relative Orientierung zur initialen Rotation interpretiert werden.

Wird die Rotation des Interaktionsgeräts jedes mal aufs neue initialisiert, so ist die Differenz zwischen der tatsächlichen Orientierung des Geräts und der vom System angenommenen arbiträr und kann auch von einem Selektionsvorgang zum nächsten wechseln. Ein intuitives und sofort sichtbares grafisches Echo des Selektionsstrahls im Blickfeld ist somit unabdingbar. Da die Pose aber immer im gleichen Maße relativ durch die Rotationssensoren gesteuert wird, besitzt das System ein jederzeit berechenbares und erwartungskonformes Verhalten.

Jeder Selektionsvorgang kann für sich als abgeschlossene Handlung betrachtet werden, die der Benutzer bewusst interaktiv einleitet und auch beendet. Die Bestimmung des Zeitpunkts kann über eine Taste am Interaktionsgerät gesteuert werden. Hier stehen wiederum zwei Alternativen der Interpretation eines Tastendrucks zur Wahl. Eine Interaktion kann durch das zweimalige Drücken der Taste begonnen und beendet werden (Abbildung 66) oder während der Dauer des Gedrückthalts der Taste aktiv sein (Abbildung 67).

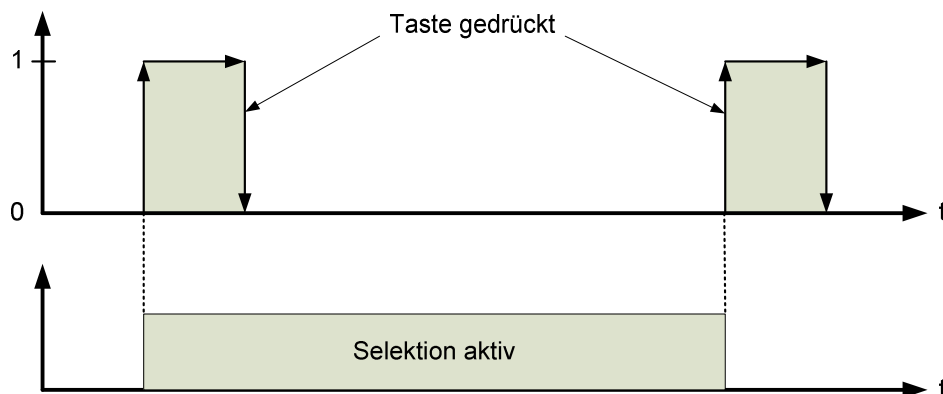


Abbildung 66: Steuerung der Selektion über ansteigende Flanken der Taste



Abbildung 67: Steuerung der Selektion über steigende und fallende Flanke des Tastenimpulses

Eine prototypische Realisierung wurde basierend auf einer Zweitasten-Maus mit einem Mausekball und einem Intersense InertiaCube2 Inertialsensor erstellt. Das Gerät kann allein mit der dominanten Hand bedient werden und ist aufgrund geringer Größe und Gewicht gut geeignet für den mobilen Einsatz.

Die Tracking-Sensoren liefern die absoluten Winkel in Relation zu einer bei der Initialisierung bestimmten Ruhelage. Diese können nun entweder zu einer absoluten oder relativen Ansteuerung des Selektionsstrahls verwendet werden. Bei der absoluten Methode werden die Winkel direkt auf den Selektionsstrahl übertragen. Die Auslenkung des Selektionsstrahls entspricht genau der Rotation des Interaktionsgerätes.

Bei der relativen Methode wird der Rotationswinkel des Eingabegeräts als Geschwindigkeit interpretiert. Je weiter die Entfernung aus der Ruhelage ist, umso schneller wird der Selektionsstrahl in die entsprechende Richtung bewegt.

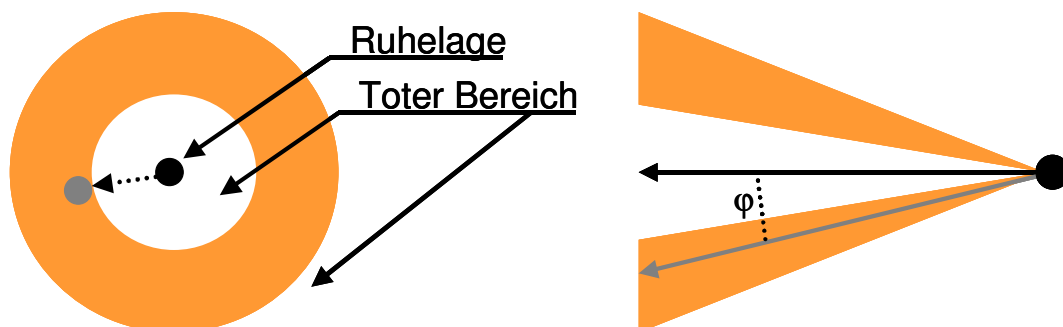


Abbildung 68: Interpretation verschiedener Auslenkungen im relativen Modus

Jede Bewegung des Interaktionsgerätes führt somit zu einer Bewegung des Selektionsstrahls oder des Cursors. Ein Verharren auf einer Position ist nur beim exakten Erreichen der Ruhelage zu erzielen. Um dieses Verhalten zu kompensieren, wird ein so genannter toter Bereich um die Ruhelage herum definiert (Abbildung 68). Dies geschieht ähnlich den Einstellmöglichkeiten bei analogen Eingabegeräten wie Joysticks. Winkelauslenkungen unter einem bestimmten Schwellwert werden dabei herausgefiltert. Erst wenn der Schwellwert überschritten ist, beginnt die Interpretation des Winkels als Geschwindigkeit. Am äußeren Ende ist dieser Wertebereich wiederum begrenzt durch einen weiteren

toten Bereich, in dem ein maximaler Geschwindigkeitswert Gültigkeit besitzt. Durch den Einsatz eines oberen Schwellwertes kann erreicht werden, dass der Selektionsstrahl nicht mit zu großer Geschwindigkeit unkontrolliert bewegt wird, sondern eine maximale Grenzgeschwindigkeit besitzt, die unabhängig von der Eingabe des Benutzers nicht überschritten werden kann. Das resultierende Mapping von Rotationswinkel (φ) des Eingabegeräts und der dazugehörigen Geschwindigkeit (V) kann Abbildung 69 entnommen werden.

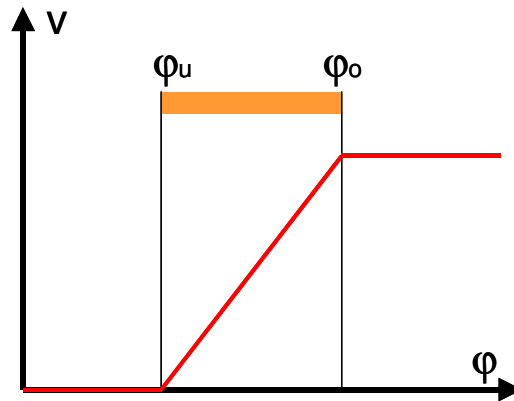


Abbildung 69: Mapping von Winkel auf Geschwindigkeit

Für die drei Bereiche, die durch die Schwellwerte φ_u und φ_o definiert werden, sehen die entsprechenden Abbildungen für die horizontalen oder vertikalen Bewegungen folgendermaßen aus:

$$\begin{aligned} \varphi < \varphi_u & : & v &= 0 \\ \varphi_u \leq \varphi \leq \varphi_o & : & v &= (\varphi - \varphi_u) * k \\ \varphi > \varphi_o & : & v &= (\varphi_o - \varphi_u) * k \end{aligned}$$

mit k = konstanter Abbildungsfaktor.

Im Bereich zwischen φ_u und φ_o wird die Geschwindigkeit linear aus den Rotationswinkeln berechnet, außerhalb des Intervalls sind die Werte konstant 0 oder entsprechen der maximalen Geschwindigkeit $(\varphi_o - \varphi_u) * k$.

Neben den Rotationen, die eine Auslenkung des Selektionsstrahls nach oben oder zur Seite steuern, ist zusätzlich ein weiterer Freiheitsgrad verfügbar. Hierbei handelt es sich um die Rotation um den Selektionsstrahl. Während diese zusätzlichen Informationen bei der Selektion in der virtuellen 3D Szene keinen Vorteil bringen, können sie bei der Manipulation eines Objekts, Menüselektion oder bei der Eingabe oder Veränderung numerischer Werte, der Quantifizierung, eine erhebliche Verbesserung der Performanz bewirken. Speziell die Menüselektion und die Quantifizierung lassen sich dabei auf ein eindimensionales Problem zurückführen, bei dem der Wert eines Parameters entweder in diskreten Schritten oder kontinuierlich geändert wird. Da nur eine einzige Rotationsachse benötigt wird, kann diese arbiträr im Raum platziert werden. Es ist allerdings darauf zu achten, dass für die größtmögliche Reiz-Reaktions-Korrespondenz eine entsprechende Visualisierung zu wählen ist (siehe Abbildung 70).

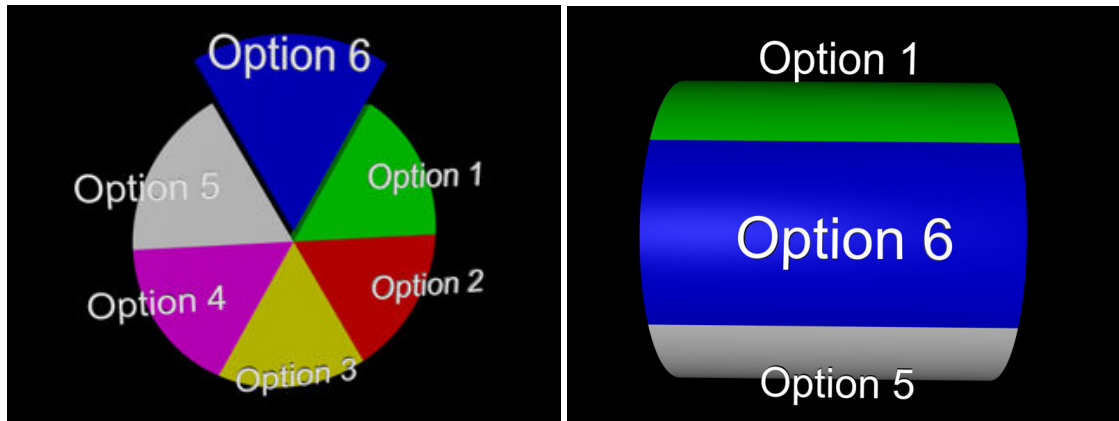


Abbildung 70: Alternative Menu-Darstellungen für unterschiedliche Rotationsachsen

Neben der sehr hohen Reiz-Reaktions-Korrespondenz, die somit erzielt werden kann ist ein weiterer Vorteil, dass nur eine relative Positionierung vorgenommen werden muss. Der Fokus wechselt bei der Überschreitung eines Schwellwertes φ_u von einem Menüeintrag zum nächsten und verharrt dort für einen festgelegten Zeitraum t . Erst nach Ablauf dieser Zeitspanne wird, wenn der Schwellwert φ_u immer noch überschritten ist ein weiterer Wechsel zum nächsten Menüeintrag ausgelöst. Dies ermöglicht, beim Erreichen des Zieleintrags, das Eingabegerät wieder in die Ruhestellung zu bringen. Wird dagegen ein weiterer Schwellwert φ_o überschritten, so wird ein kontinuierlicher Wechsel der Menüeinträge in der entsprechenden Drehrichtung ausgelöst. Dies dient der schnellen Überbrückung größerer Distanzen innerhalb des Menüs.

$$\begin{aligned} \varphi < \varphi_u & : & m_i &= m_i \\ \varphi_u \leq \varphi \leq \varphi_o & : & m_j &= m_{i+1} \text{ oder } m_j = m_{i-1} \\ \varphi > \varphi_o & : & m_j &= m_{i+k} \text{ oder } m_j = m_{i-k} \end{aligned}$$

Während der Interaktion werden nur Rotationen um eine der Hauptachsen des Eingabegerätes ausgewertet, die anderen Rotationsrichtungen werden während der Menüselektion ignoriert. Das resultierende Interaktionsgerät bietet die gewohnte Tastenfunktionalität einer Desktopmaus kombiniert mit der beschriebenen Realisierung der 3D Selektion über einen Inertial-Sensor und den spezialisierten Interaktionsmodi für die Bedienung von Menüs.

Verglichen mit dem iOrb besteht der Unterschied vor allem im direkteren Mapping der Eingabebewegung auf das Verhalten des Interaktionsverfahrens, speziell bei der Selektion in 3D. Das Ergebnis zeigt sich hierbei in einer wesentlich stärkeren Reiz-Reaktions-Korrespondenz. Bei der Bedienung von Menüs werden der Bedienmodus und die Interpretation der Rotationen geändert. Das Verhalten ist dabei allerdings so intuitiv und erwartungskonform, dass es vom Anwender sofort erfasst werden kann.

Werden über die Menüselektion hinausgehende, komplexere Eingaben vom Benutzer gefordert, sind diese Aufgaben nur auf relativ kompliziertem Weg mit dem oben beschriebenen Interaktionsgerät zu bewerkstelligen. Komplexere Aufgaben müssen auf viele einfache Interaktionsaufgaben, die hintereinander durchzuführen sind, herunter gebrochen werden. Neben dem dadurch steigenden Bedarf an Bearbeitungszeit, geht auch die Übersicht über die Gesamtaufgabe verloren.

Im Folgenden wird deshalb ein Interaktionsgerät vorgestellt, das neben der schon beschriebenen Strahlselektion mittels Rotationssensoren auch die 2D Eingabe mittels Stift-Metapher erlaubt. Das resultierende Eingabegerät kann somit als Adaption des PIP-Konzepts angesehen werden, basierend auf der beschriebenen Kombination von Inertial-Sensor zur Bestimmung der Orientierung und einem Benutzermodell zur Bestimmung der Position des Tablett. Das Tablett kann sich somit außerhalb des Blickfelds des Anwenders befinden und in einer natürlichen Armhaltung bedient werden.

Im Gegensatz zur Kombination von Maus und Inertial-Sensor wird das Tablett mit der nicht-dominanten Hand gehalten während der Stift mit der dominanten Hand geführt wird. Da die Selektion über die Rotation des Tablett gesteuert wird, erfolgt diese hier über die nicht-dominante Hand. Diesem Nachteil bei der Selektion steht der Vorteil der zusätzlichen Interaktionsmöglichkeiten über den Stift gegenüber. Da sich das Tablett nicht im Blickfeld des Anwenders befindet, wird hier die Propriozeption des Anwenders ausgenutzt. Über das Selbstgefühl kann dieser sehr gut abschätzen, wo sich der Stift relativ zum Tablett befindet. Zusätzliche Unterstützung erhält er dabei über das taktile Feedback des Stifts auf der Oberfläche des Tablett und ein entsprechendes optisches Feedback im HMD.

Mit einem Grafiktablett alleine ist der Benutzer in der Lage den Cursor absolut in 2D zu positionieren. Kombiniert man das Tablett nun noch mit einem Inertialsensor, sind auch rotatorische 3D Interaktionen möglich. Im Folgenden sollen nun die mit diesem Gerät realisierten Eingabetechniken und –metaphern vorgestellt werden.

Das Grafik-Tablett erlaubt es zu erkennen, wenn der Stift sich in Interaktionsreichweite über dem Tablett befindet. Diese Eigenschaft ermöglicht es, die damit verknüpften Interaktionsmechanismen on demand zu aktivieren. Eine mit dem Tablett verknüpfte GUI wird erst durch das Aufsetzen des Stifts aktiviert und kann wieder deaktiviert werden, sobald der Stift entfernt wurde. Da eine solche GUI nur sporadisch erscheint und dazu auch bewusst vom Anwender aktiviert wird, ist es nicht so störend, dass große Teile des Blickfelds von der GUI eingenommen werden.

Im einfachsten Fall wird eine head-stabilized Benutzeroberfläche dargestellt, die die gerade relevanten Bedienelemente enthält.

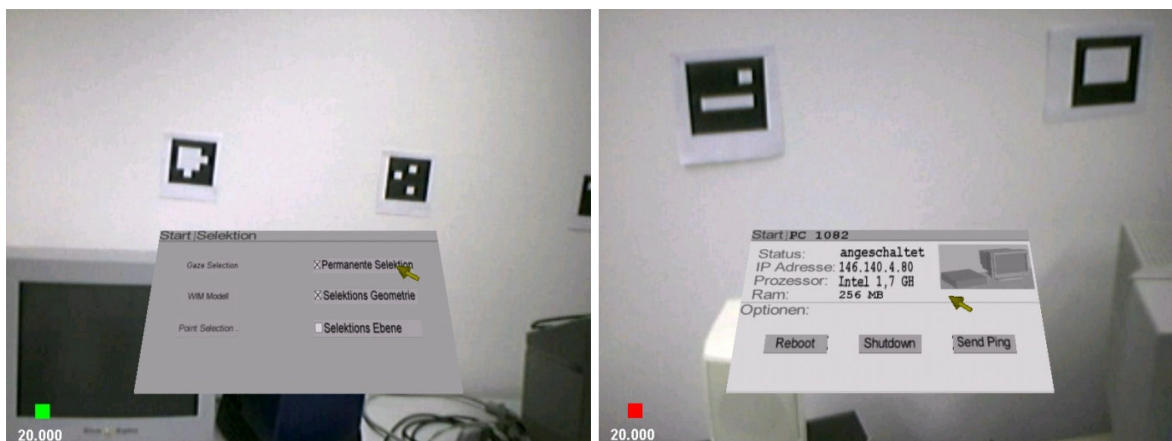


Abbildung 71: Head-Stabilized GUI in Verbindung mit Tablett

Eine zusätzliche Hilfe bei der gezielten Platzierung des Stifts auf dem Tablett stellt ein geeignetes grafisches Feedback dar, wie der Pfeilförmige Cursor in Abbildung 71. Da die Ansteuerung eines Or-

tes auf der virtuellen GUI sehr präzise vorgenommen werden kann, lassen sich so auch komplexe GUIs steuern oder sogar Texte eingeben und Zeichnungen anfertigen. Auch hier ist gegenüber dem iOrb Konzept ein klarer Bruch im verwendeten Interaktionsmodus zu erkennen, wodurch allerdings erst eine große Zahl an Anwendungsbereichen erschlossen werden kann. Die Interaktion über die zusätzliche 2D Eingabe lässt sich wesentlich universeller nutzen, als das doch sehr limitierende Konzept der Eingabe allein über Rotationen. Da der Arbeitsbereich der GUI exakt mit der Eingabefläche des Tablets korrespondiert, kann auch für die 2D Eingabe eine hohe Reiz-Reaktion-Korrespondenz erreicht werden kann.

5.5 GUI Darstellung in Augmented Reality

Obwohl klassische grafische Benutzerschnittstellen (GUIs) in Augmented Reality eine weitaus geringere Rolle spielen als in einer WIMP Desktop Oberfläche, gibt es Operationen die über eine zweidimensionale GUI wesentlich leichter und präziser durchzuführen sind, als über Interaktionen in der virtuellen 3D Umgebung [LIND99]. Ein Beispiel hierfür ist die Einstellung eines skalaren Wertes.

Obwohl Augmented Reality ein neue Art der Benutzerschnittstelle darstellt, die völlig neuartige Interaktionstechniken ermöglicht, ist es oftmals effizienter, auf klassische Konzepte, die ursprünglich für WIMP-Benutzerschnittstellen entwickelt wurden, zurückzugreifen.

Voraussetzung dafür, dass solche virtuelle GUIs in der 3D Umgebung wirklich Effizienzvorteile bringen ist, dass die GUI und die verwendeten Interaktionstechniken entsprechend aufeinander abgestimmt sind. Die Bedienung eines Sliders auf der GUI über eine 3D Selektion und anschließende Einstellung des Zielwertes mittels Translation oder Rotation des Eingabegeräts sind wesentlich komplizierter und zeitaufwändiger als die Bedienung eines Sliders über ein Grafiktablett und das präzise Wählen des Zielwertes mit der Stift-Metapher. Hieraus lässt sich ableiten, dass die Auswahl an die Aufgabe angepasster Interaktionsgeräte entscheidenden Einfluss auf die Performanz des Benutzers bei der Durchführung einer spezifischen Aufgabe hat und die GUI Komponente die Verwendung unterschiedlicher Interaktionsgeräte optimal unterstützen muss. So sieht die optimale Darstellung einer GUI, die mit 3D Selektion bedient wird, grundlegend anders aus, als eine grafische Benutzerschnittstelle, die mit einem Stift und einem Grafiktablett bedient wird.

Grundsätzlich können GUI-Elemente, ebenso wie andere Darstellungselemente sowohl world-stabilized als auch display-stabilized dargestellt werden. Im Falle einer hybriden GUI, bei der die grafische Benutzerschnittstelle auf einem separaten Rechner dargestellt wird, ist eine GUI gleicher Funktionalität in einer 2D-Umgebung zu erzeugen.

Grundsätzlich kann zwischen zwei Arten der Erzeugung solcher GUIs unterschieden werden. Als erstes wären hier die bereits in einem vorgelagerten Authoringschritt erstellten GUIs zu nennen. Hierbei handelt es sich um applikationsspezifische Elemente, die speziell mit der Applikationslogik und den anderen Präsentationsdaten verbunden sind. Als Beispiel hierfür kann die Steuerung der Darstellung einer interaktiven Produktpräsentation oder von Rekonstruktionen antiker Artefakte genommen werden. Über die GUI-Elemente kann der Benutzer intuitiv Oberflächeneigenschaften von Objekten ändern oder Aktionen auslösen, die eine bestimmte Funktion erklären.

Bei der anderen Variante wird die GUI dagegen on the fly zur Laufzeit generiert. Die Erzeugung geschieht dabei aus einem Satz gegebener Informationsobjekte, denen auf Basis ihres Typs spezifische GUI-Elemente zugeordnet werden. Handelt es sich dabei um eine Menge von Informationsobjekten kann automatisch ein Menü zur Auswahl der gewünschten Informationen generiert werden. Bei einem einzelnen Objekt hingegen können entsprechende Controls direkt erzeugt und angezeigt werden, die an den jeweiligen Objekttyp angepasst sind.

Die verwendeten GUI Widgets sollten optimal auf die jeweils verwendete Display- und Interaktionstechnik abgestimmt sein. Bei den Displays sind die Hauptfaktoren, ob es sich um ein HMD oder ein handgeführtes Display handelt, welche Auflösungen und wie viele Farben die Darstellung unterstützt. Beim Interaktionsgerät ist eine genaue Anpassung an dessen Funktionalitäten unter der besonderen Berücksichtigung seiner Stärken und Schwächen notwendig.

Die Problemstellung ist eine ähnliche, wie sie sich für Entwickler von Benutzerschnittstellen für mobile Endgeräte, wie PDAs und Smart-Phones stellt. Auch hier sind unterschiedliche Displays und Eingabemethoden zu unterstützen. Der zeitaufwändige Vorgang der Erstellung von Benutzerschnittstellen muss dabei für unterschiedliche Plattformen wiederholt durchgeführt werden.

Die Lösung dieser Problematik stellt die Abstraktion von Benutzerschnittstellen über entsprechende abstrakte Beschreibungssprachen dar. Je nach Konfiguration und Parametrierung des Systems können aus einer einzelnen abstrakten Beschreibung unterschiedliche grafische oder auch nicht grafische Repräsentierungen erzeugt werden.

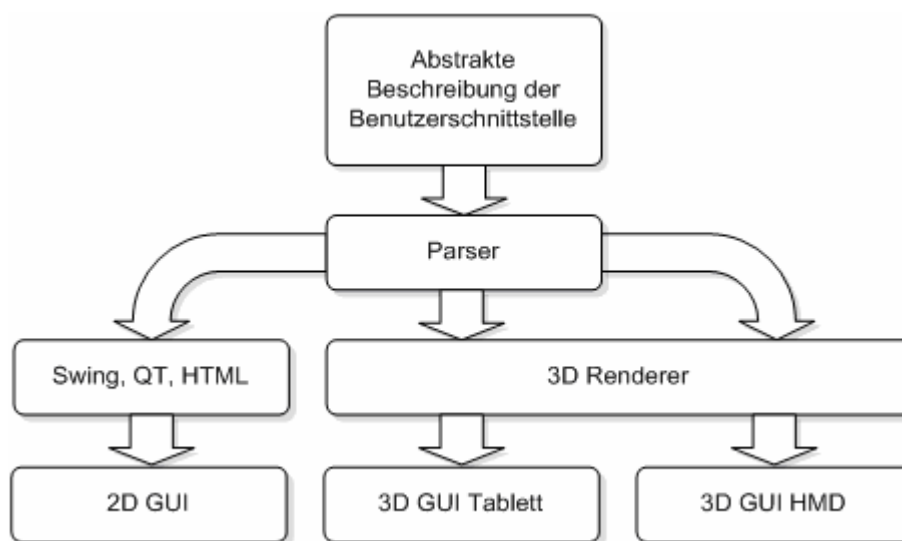


Abbildung 72: Generierung verschiedener Darstellungen aus einer abstrakten Beschreibung

5.5.1 GUI-Abstraktion

Die Abstraktion von Benutzerschnittstellen erlaubt das einmalige Erstellen einer Benutzerschnittstellen-Beschreibung, aus der zur Laufzeit gerätespezifische Benutzerschnittstellen erzeugt werden können. Die aufwändige Erstellung von Benutzerschnittstellen muss somit nur einmal durchgeführt werden. Das Ergebnis des Erstellungsprozesses kann in verschiedenen Konfigurationen wieder verwendet werden. Der Aufwand, der für diese Flexibilität getrieben werden muss, besteht vor allem in einer

zusätzliche Komponente des Laufzeitsystems, die aus der abstrakten Beschreibung die konkrete Darstellung der Benutzerschnittstelle erzeugt. Die Funktionalität dieser Komponente lässt sich aufteilen in das Parsen der Beschreibung und die Erstellung der GUI.

Bisher haben sich vor allem zwei konkurrierende Ansätze der GUI-Abstraktion herauskristallisiert, UIML (User Interface Markup Language) und XIML (Extensible Interface Markup Language).



Abbildung 73: Userinterface Generierung aus abstrakter Beschreibung

Da kein klarer Trend bezüglich eines zukünftigen Standards erkennbar ist, wurde die hier vorgestellte Architektur so ausgelegt, dass die Parser für unterschiedliche Beschreibungssprachen losgelöst von der eigentlichen GUI-Komponente realisiert sind. So ist es möglich, auch andere Beschreibungssprachen zu unterstützen.

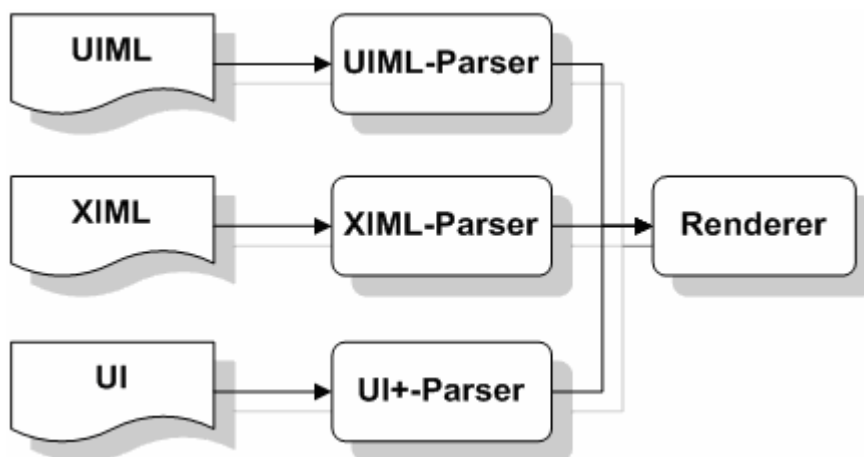


Abbildung 74: Unterstützung verschiedener GUI Beschreibungssprachen

Ein entscheidender Faktor für die Umsetzung des Konzepts ist das Vorhandensein einer funktionierenden Toolkette. Neben den Laufzeit-Komponenten kommt hier vor allem den Werkzeugen zur Erstellung der GUI-Beschreibung besondere Bedeutung zu. Da Anwendungs-Entwickler hier meist komfortable Werkzeuge gewohnt sind, sollte eine Umgebung zur Erzeugung abstrakter Benutzerschnittstellen ähnlichen Ansprüchen genügen. Aus diesem Grund wurde eine Anbindung für UI, die GUI Beschreibungssprache des GUI-Toolkits Qt von Trolltech entwickelt. In diesem Toolkit existiert ein Werkzeug, der Qt-Designer, das die komfortable Erstellung auch komplexer Benutzerschnittstellen ermöglicht.

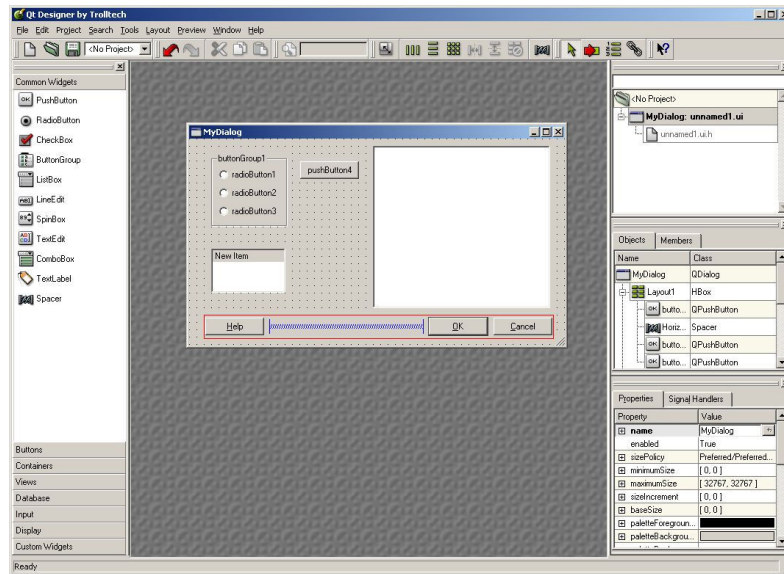


Abbildung 75: QT Designer [Trolltech]

Wie die wirklich abstrakten Beschreibungssprachen, werden auch die QT spezifischen UI Beschreibungen in einer XML Datei abgelegt. Diese Beschreibung enthält die hierarchische Beschreibung der GUI-Elemente, ihrer räumlichen Anordnung und ihrer spezifischen Eigenschaften, wie etwa Text und Größe. Um das Konzept der GUI-Abstraktion nun in VR/AR-Anwendungen nutzen zu können, ist die Generierung von grafischen Benutzerschnittstellen aus den Beschreibungen notwendig.

In einem Szenengraphbasierten Rendering-System, wie es dem AR-Viewer zu Grunde liegt, bietet es sich an, die GUI aus Szenengraph-Knoten aufzubauen. Eine so erzeugte GUI lässt sich, wie in Abschnitt 4.4.1 beschrieben, in den verschiedenen Darstellungsmodi, angepasst an die jeweilige Situation, präsentieren. Die Komponente des Renderers in Abbildung 74 ist somit zu ersetzen durch eine Komponente, die aus der Userinterfacebeschreibung entsprechende Szenengraphknoten und Teilbäume aufbaut. Im Folgenden wird diese Komponente GUI-Manager genannt.

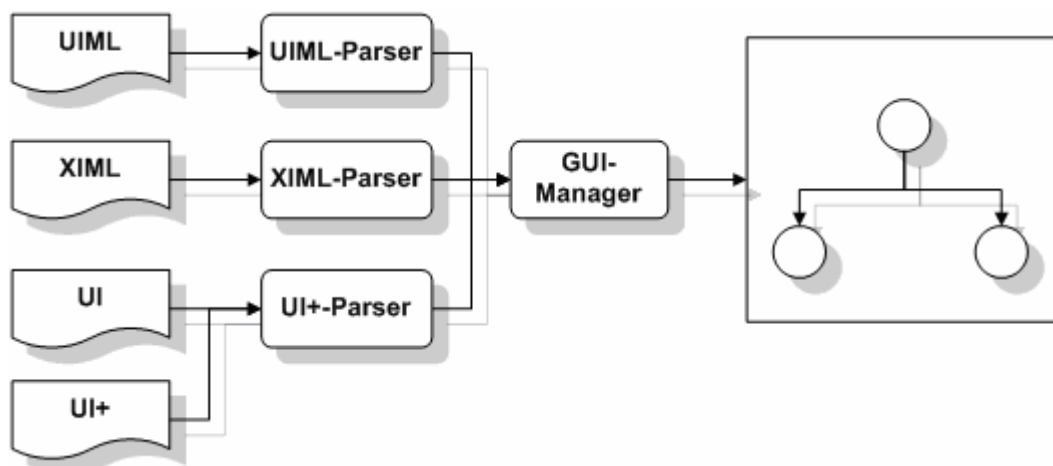


Abbildung 76: Laufzeitsystem zur Erzeugung von 3D GUIs

5.5.2 GUI-Manager

Die Komponente des Userinterface-Managers hält den Status der grafischen Benutzerschnittstelle. Es werden von allen aktuell sichtbaren GUI-Elementen die Eigenschaften und der Status gespeichert. Die Speicherung geschieht hier in einer hierarchischen Baumstruktur.

Diese abstrakten GUI-Objekte besitzen einen Verweis auf die ihnen zugeordneten sichtbaren GUI-Knoten im Szenengraphen. Dabei können einem GUI-Objekt durchaus mehrere zusammenhängende Knoten, also Teilbäume des Szenengraphen zugeordnet werden. Zusammengefasst besteht ein GUI-Element also aus einem GUI-Objekt im Manager und den Knoten im Szenengraphen.

5.5.3 GUI Objekte

In den Knoten des GUI-Graphen ist der Zustand der UI-Elemente gespeichert. Ändert sich dieser Zustand, veranlasst das Objekt selbst die Änderung an der graphischen Repräsentation. Jedes Objekt ist selbst für die Verwaltung der ihm zugehörigen Knoten im Rendergraphen zuständig. Für das Erzeugen und das Entfernen von Knoten ebenso wie für die Reaktion auf Zustandsänderungen. Wird ein GUI-Objekt erzeugt und in den GUI-Graph gehängt, erzeugt es die grafische Darstellung in Form von einem oder mehreren Szenengraph-Knoten. Ein Button kann durchaus von einem einzelnen Geometrie-Knoten repräsentiert werden, ein Slider besteht aber aus mindestens zwei Geometrie-Knoten, von denen einer relativ zum anderen transliert werden kann.

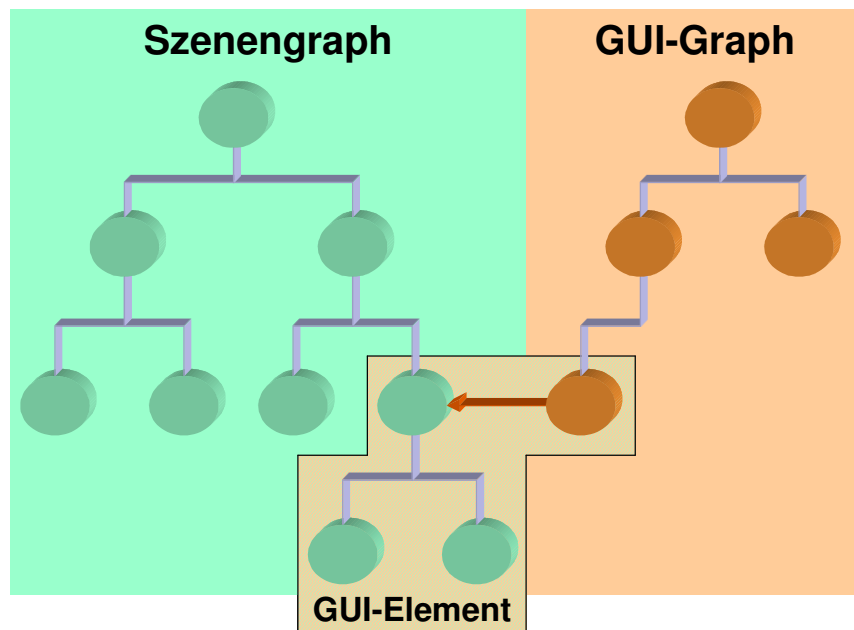


Abbildung 77: Struktur eines GUI-Elements

Die Funktionalität eines GUI-Elements ist vollkommen unabhängig von seiner graphischen Repräsentation. Jedes Element des Userinterfaces besitzt genau ein Attribut, dessen Wert der Benutzer durch Interaktion mit dem GUI-Element verändern kann. Das Attribut signalisiert dabei den veränderlichen Zustand des GUI-Elements.

Die graphische Repräsentation eines GUI-Knotens kann in jedem Teil des Szenengraphen generiert werden, also sowohl im orthografischen Viewport, als auch in der AR Szene. Ein GUI-Element kann somit sowohl zwei- als auch dreidimensional dargestellt werden. Zusätzlich können Knoten dynamisch zwischen den Teilgraphen verschoben werden, so dass die Konfiguration des GUI-Managers auch on the fly geändert werden kann.

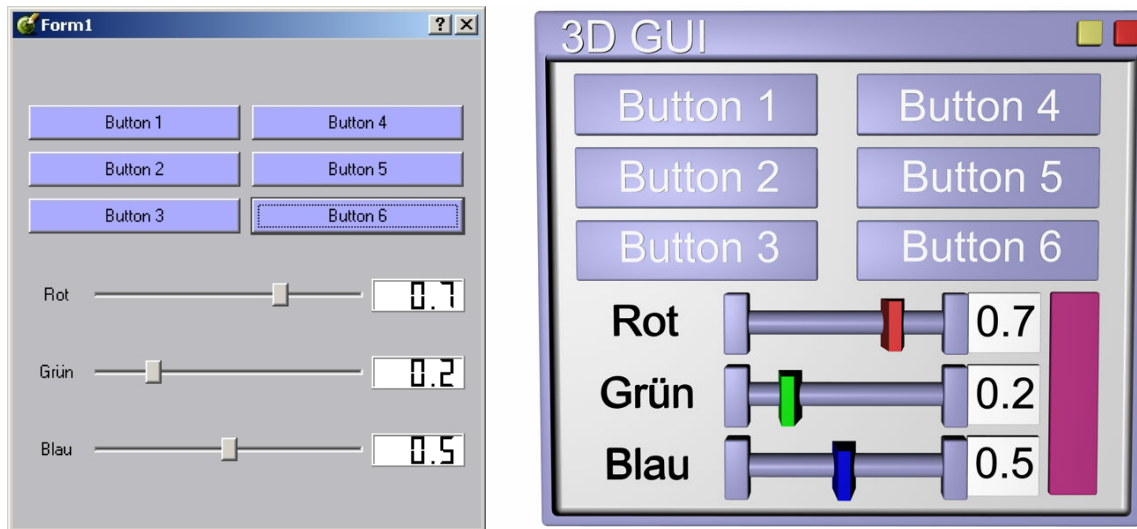


Abbildung 78: Qt GUI und entsprechende 3D Repräsentation

Im Umfeld von VR und AR spielt die Anpassung der GUI-Darstellung an die jeweiligen Gegebenheiten eine wichtige Rolle. Bei den Einflussfaktoren, die hierbei neben den besonderen Bedürfnissen und Vorlieben des Anwenders zu beachten sind, handelt es sich vor allem um die technischen Randbedingungen, die durch die verwendeten Ein- und Ausgabegeräte vorgegeben werden. Zusammenfassen lässt sich dies in Benutzer- und Hardwareprofilen, die diese Vorgaben für jeweils eine Konfiguration beschreiben. Bei der Instanziierung der GUI-Elemente aus einer abstrakten Beschreibung beeinflussen die Profile, wie die jeweilige konkrete GUI-Darstellung aussieht. Dabei kann beeinflusst werden, welche GUI-Elemente erzeugt werden, also beispielsweise welches Control für die Einstellung eines skalaren Wertes erzeugt wird, und über das Setzen von Attributen der Controls auch deren Darstellung und Verhalten parametrisiert werden. Beispiele für solche Attribute sind die Schrittweite eines Sliders, wie dieser beschriftet werden soll und ob er horizontal oder vertikal darzustellen ist.

Zusätzlich unterstützt das hier umgesetzte Konzept zur Generierung von Benutzerschnittstellen auch die Verwendung unterschiedlicher Darstellungsstile, im Weiteren Style genannt. Ein Style umfasst dabei einen kompletten Satz an Beschreibungen der GUI-Geometrien. Jede Geometriebeschreibung entspricht dabei einem GUI-Element Typen und kann somit auch mehrere Teilgeometrien beinhalten.

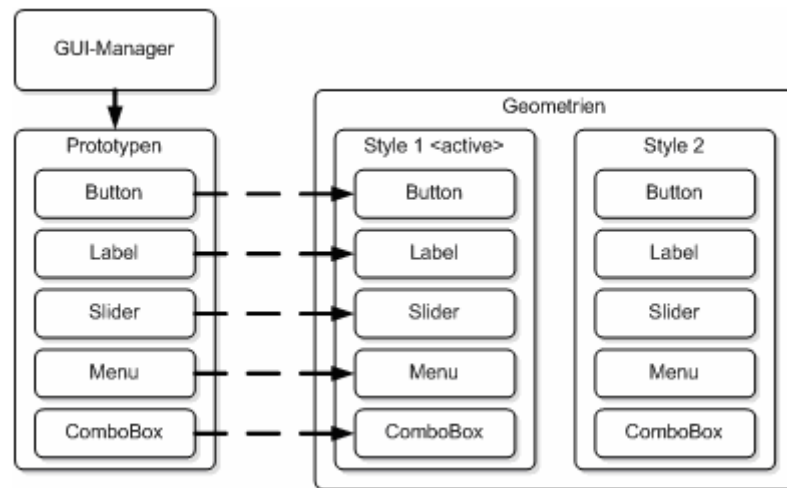


Abbildung 79: Beziehung zwischen Prototypen, Geometrien und Styles

Damit das Zusammenspiel der verschiedenen GUI Elemente gewährleistet werden kann, müssen die einzelnen Elemente gewisse Regeln bezüglich ihrer geometrischen Eckdaten und der Benennung einhalten, da die Referenzierung der Felder eines Elements zur Laufzeit über deren Namen erfolgt. Die geometrischen Eigenschaften eines Elements beeinflussen dessen Verhalten bei der Erzeugung einer aus mehreren GUI-Elementen zusammengesetzten, komplexen GUI. Hier hilft ein Layout-Manager aus den einzelnen Elementen ein optisch und funktional ansprechendes Ganzes zu erzeugen. Angesteuert wird der GUI-Manager durch spezielle Attribute der GUI-Elemente, aus denen sich zusammen mit der Reihenfolge, in der die Elemente in der Beschreibung auftauchen, deren Nachbarschaftsbeziehungen aufbauen lassen.

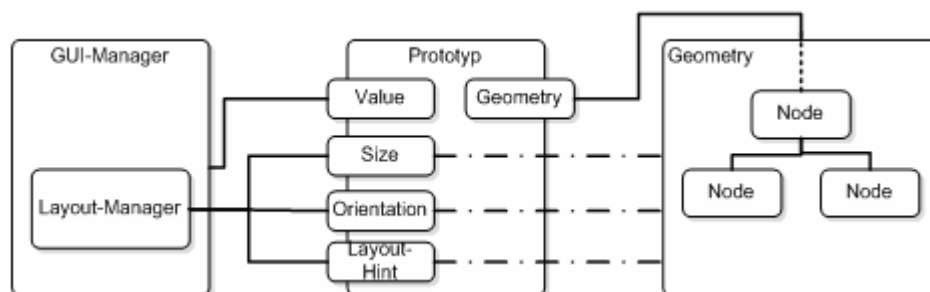


Abbildung 80: Zusammenwirken zwischen Layout-Manager und GUI-Element

Die eigentliche Funktionalität eines GUI Elements befindet sich in parametrierbaren Prototypen, die wiederum auf eine Geometrie verweisen. In der vorliegenden Realisierung sind diese Prototypen und Geometrien in VRML, einer Beschreibungssprache, die ursprünglich für webbasierte 3D-Welten konzipiert wurde, abgelegt. Die Prototypen beinhalten die Verbindungslogik in Form von Scripten und mindestens einen Sensorknoten zur Erzeugung der notwendigen Interaktions-Events. Die Verbindung zwischen diesen GUI Elementknoten und den dazugehörigen Geometrieknoten erfolgt zum einen über das Einhängen des Geometrieknotens in den GUI Knoten und zum anderen durch das Erzeugen entsprechender Routes zwischen den Feldern von Sensor, Script und Geometrie.



Abbildung 81: 3D GUI im AR-Browser

Bei der Instanziierung eines GUI-Elements sind mehrere Phasen zu durchlaufen. Nach dem Aufbau der internen Strukturen eine GUI Knotens wird die Geometrie der Verwaltung durch den Layout-Manager übergeben, der die entsprechenden Felder des GUI-Knotens ausliest und abschließend die Geometrien der zusammengehörigen GUI-Knoten entsprechend ansteuert und platziert. Abschließend werden die externen Verbindungen aufgebaut.

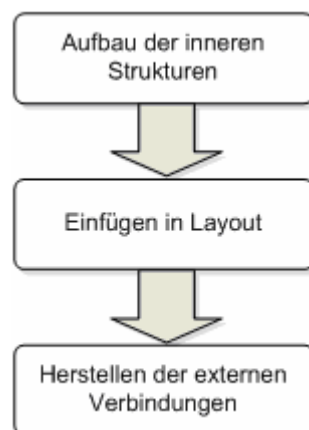


Abbildung 82: Ablauf der Instanziierung eines GUI-Elements

Das Value-Feld des GUI-Knotens wird mit dem entsprechenden Zielfeld verbunden, wenn dieses über seinen Namen gefunden werden kann. Bei dem Zielfeld kann es sich im einfachsten Fall um ein ande-

res Feld eines Szenengraphen-Knotens handeln, so dass ein Attribut dieses Knotens interaktiv gesteuert werden kann. Um die GUI-Elemente jedoch universeller einsetzen zu können ist es notwendig, dass auch Attribute außerhalb des Szenengraphen damit verbunden werden können. Hierbei kann es sich um beliebige Attribute von Software-Modulen des AR Browsers handeln um das Verhalten des AR Browsers damit zu steuern oder um externe Verbindungen über das Scripting Interface des AR Browsers um somit externe Prozessverbindungen herzustellen. Da die Verbindungen bi-direktional sind, also das anzusteuern Attribute auch den Zustand des GUI-Controls ändern kann, ist es möglich, dass ein GUI Element in der AR Szene sowohl für die Visualisierung als auch für die Steuerung externer Prozesse verwendet werden kann.

Neben der bereits beschriebenen Möglichkeit der Erzeugung einer GUI aus einer Beschreibungsdatei ist auch die Generierung zur Laufzeit aus einem gegebenen Satz von Attributen von entscheidender Bedeutung für die universelle Verwendbarkeit des Systems. Nach der Bestimmung der Typen der anzusteuern Attribute und Felder können die dafür passenden GUI Elemente erzeugt und damit verknüpft werden. Das gewünschte Mapping zwischen Attribut-Typen und den damit zu verknüpfenden Typen von GUI Elementen kann anhand einer Abbildungstabelle a priori festgelegt werden.

5.6 Zusammenfassung

Da in Augmented Reality die Umgebung des Benutzers zum Interface zu den damit verknüpften Daten wird, kommt dem Thema Interaktion besondere Bedeutung zu. Bei der Betrachtung existierender Interaktionsgeräte für immersive VR Umgebungen und Eingabegeräten für mobile Anwendungen, fällt auf, dass diese Geräte nicht den Anforderungen an ein Eingabegerät für mobile AR Anwendungen entsprechen. Entweder sie unterstützen 3D Interaktionen, wie die Selektion im Raum, erfordern dafür aber eine Tracking-Infrastruktur, wie sie im mobilen Umfeld nicht bereitgestellt werden kann. Oder aber sie sind mobil einsetzbar, lassen sich aber nicht für die 3D Interaktion verwenden.

Dieser Mangel führt zur Konzeption des universellen Eingabegerätes zur Interaktion in mobilen AR Umgebungen. Ein Gerät, dass mit Hilfe speziell darauf abgestimmter Interaktionstechniken sowohl die Selektion in 3D erlaubt, als auch die Bedienung von 2D Benutzeroberflächen unterstützt.

Besonders gut zur Geltung kommen die Stärken des Interaktionsgerätes in Verbindung mit den hier vorgestellten Techniken zur Generierung von 2D Benutzerschnittstellen innerhalb der 3D Umgebung. Während die Selektion eines Objekts im Raum über die 3D Selektionstechniken mittels Raycasting geschieht, werden die Elemente der mit dem Objekt verknüpften GUI über die Stift und Tablett Metapher bedient.

Neben diesen Stärken im Zusammenspiel mit dem universellen Interaktionsgerät erlaubt das GUI Toolkit sowohl die Generierung von Benutzerschnittstellen aus abstrakten GUI Beschreibungen als auch die on-the-fly Erzeugung aus einem Satz von Attributen, die über diese GUI beeinflusst werden können. Die Verwendung des frameworkweiten Attributmechanismus erlaubt es, über solche GUIs alle Informationsobjekte und sogar die Komponenten des AR-Systems selbst zu bedienen.

6 Engineering mobiler AR Anwendungen

6.1 Engineeringaufgaben

6.1.1 Einrichten und Kalibrieren von Tracking-Systemen

Der Aufwand für das Installieren und Einrichten des Tracking-Systems ist stark vom verwendeten Tracking-Verfahren abhängig. So müssen bei markerbasiertem Video-Tracking Marker in der Arbeitsumgebung angebracht und ein entsprechendes Modell der Tracking-Informationen im Rechner modelliert werden. Zusätzlich muss die absolute Lage der Marker bezüglich des Koordinatensystems der Anwendung bestimmt werden. Der übliche Weg ist hierbei, dass die Marker in der realen Umgebung angebracht werden und anschließend jeder Marker im Rechner entsprechend modelliert wird. Die reale Position und Orientierung der Marker wird dabei vermessen und in den Rechner eingegeben. Der Aufwand für das Anbringen der Marker und das spätere Vermessen kann dabei beträchtlich sein. Neben der komplett manuellen Vermessung der Marker existieren aber bereits semiautomatische Methoden, bei denen sich ausgehend von der bekannten Position und Orientierung eines Markers weitere Marker im gleichen Koordinatensystem kalibrieren lassen [BARA02].

Andere Tracking-Verfahren erfordern an dieser Stelle natürlich auch andere Methoden der Vorbereitung und Kalibrierung. Videobasierte Tracking-Verfahren, die mit natürlichen Merkmalen der Umgebung arbeiten, erfordern die Erstellung eines Modells der Umgebung, dass Beschreibungen der Merkmale enthält, deren Erkennung im Kamerabild mit hoher Wahrscheinlichkeit erwartet werden kann. Da auf dem Gebiet der featurebasierten Tracking-Verfahren noch eine große Anzahl an Verfahren konkurriert und noch nicht absehbar ist, wie ein wirklich erfolgreiches Verfahren aussehen wird, sind auch bei der Vorbereitung solcher Verfahren noch keine endgültigen Aussagen bezüglich des Vorgehens und des daraus resultierenden Aufwands möglich.

Das Vorbereiten und Kalibrieren des Tracking-Systems muss dabei für jede Umgebung, in der das System betrieben wird, einmalig durchgeführt werden. Allerdings ist hier die Definition des Begriffs Umgebung unscharf, da beispielsweise in einer Autowerkstatt die Umgebung aus dem Auto, das repariert werden soll und der umgebenden Werkstatt besteht. Während die Werkstatt relativ invariant ist, müssen verschiedene Autotypen erwartet werden, die zudem nie an exakt der gleichen Position stehen werden. Für ein markerbasiertes Tracking-System bedeutet dies, dass sich Marker in zwei unabhängigen Koordinatensystemen befinden, das Koordinatensystem des Autos und das der Werkstatt. Hier kann allerdings leicht eine einfache Transformation von einem in das andere System gefunden werden, da es sich nur um eine Translation und Rotation handeln kann. Problematisch ist hierbei lediglich die genaue Bestimmung dieser Transformation.

6.1.2 Kalibrierung von See-Through-Displays

Ein zusätzlicher Vorbereitungsschritt entsteht bei der Verwendung eines See-Through Displays. Das Display muss sowohl relativ zum Tracking-System als auch zum Benutzer selbst kalibriert werden. Hierfür existieren unterschiedliche Methoden, von denen der SPAAM Algorithmus wohl der am weitesten verbreitete ist [TUCE00]. Dieser Algorithmus setzt voraus dass der Benutzer für mehrere Punkte interaktiv eine 2D-3D-Korrespondenz herstellt. Aus diesen Korrespondenzen kann das System dann die entsprechende Transformation und Projektion berechnen.

Eine vollständige Kalibrierung ist nur notwendig, wenn ein neuer Benutzer das Display verwendet. Eine Korrektur des Kalibrierungsergebnisses ist allerdings bei jedem Aufsetzen des Displays notwendig, da die relative Position des Displays zum Auge des Benutzers nie exakt gleich ist. Es ist also sinnvoll diese Kalibrierungsfunktionalität in das eigentliche Augmented Reality Laufzeitsystem zu integrieren, so dass der Benutzer im Falle einer De-Kalibrierung jederzeit Korrekturen vornehmen kann.

6.1.3 Modellierung des virtuellen Informationsraums

Die Modellierung der Informationen ist für jede Anwendung einmalig durchzuführen. Von entscheidendem Einfluss auf die Qualität der jeweiligen AR Anwendung sind die Daten, die dem Benutzer präsentiert werden. Das Paradigma der kontextabhängigen Informationspräsentation erfordert hier, dass dem Benutzer die jeweils ‚richtigen‘ Informationen in Bezug auf seinen Ort, die Zeit und seine Absichten bereitgestellt werden können. Für den Fall des Service-Technikers bedeutet dies, dass ihm zum gerade aktuellen Arbeitsschritt die erforderlichen Anweisungen präsentiert werden. Zusätzlich müssen diese Informationen den Bedürfnissen und Vorlieben des jeweiligen Benutzers angepasst sein. Dies betrifft vor allem die Art der Präsentation und den Detaillierungsgrad der Informationen.

Auf der Seite des Autors der Service-Anleitung, eines technischen Redakteurs, ist das Vorgehen bei der Erstellung vor allem abhängig von der Art, der für ihn verfügbaren Information. Die Spanne reicht hier von der rein textuellen Beschreibung der durchzuführenden Arbeitsschritte über zweidimensionale Darstellungen wie Illustrationen, CAD-Zeichnungen und Pläne bis zu detaillierten 3D-Modellen. Die notwendigen Schritte aus diesen Informationen Beschreibungen von Arbeitsschritten zu erzeugen, können demzufolge sehr unterschiedlich ausfallen. Dies spiegelt sich auch in der Ausgestaltung von Autorenwerkzeugen wieder, wie die folgende Übersicht in der Literatur beschriebener Autorensysteme für unterschiedliche AR Anwendungen verdeutlicht.

Der AMIRE Authoring-Wizard beruht auf dem interaktiven Zusammenfügen von Bauteilen in einer Mixed Reality Umgebung [ZAUN03]. Die daraus resultierende Montageanleitung wird unter Zuhilfenahme der realen Bauteile und deren Beziehung untereinander erstellt. Die realen Bauteile werden dabei über ein markerbasiertes Tracking-Verfahren lokalisiert und identifiziert. Die Platzierung der virtuellen Geometrien relativ zu den entsprechenden realen Bauteilen wird interaktiv in der MR/AR-Sicht vorgenommen. Auch beim Tinmith-Metro System liegt der Schwerpunkt auf der Modellierung in einer AR-Umgebung [PIEK01]. Allerdings ist hier die Aufgabe die 3D- Rekonstruktion einer urbanen Umgebung zu erstellen.

Eine gänzlich andere Zielrichtung verfolgt das Tiles-System, bei dem ebenfalls in der MR/AR-Umgebung selbst die Szene erstellt wird [POUP01]. Allerdings dient hier die Szenenerstellung dem Rapid Prototyping. Der Authoringvorgang selbst ist hier die eigentliche Applikation. Ein System bei dem der Autor sich nicht selbst in einer MR/AR-Umgebung befindet, stellt das von Haringer präsentierte Konzept dar [HARI02]. Der Autor nimmt hier Teile der Szenenerstellung in der Standard Präsentationssoftware Microsoft Powerpoint vor. Die einzelnen Schritte einer Arbeitsanweisung spiegeln sich hierbei in der Aufeinanderfolge der Folien einer Präsentation wieder. Erst einem späteren Weiterverarbeitungsschritt werden die beschriebenen Objekte im 3D Raum verteilt. Die Autorenkomponente des MARS Systems erlaubt Verknüpfung von Multimediaobjekten mit realen Orten [GUEV03]. Der Redakteur erzeugt *points of interest* in einer dreidimensionalen virtuellen Umgebung und verknüpft die verschiedenen Multimedia Objekte mit Bezug zu diesem Punkt.

Die Informationen müssen in ein räumliches Modell gebracht werden, so dass sie am entsprechenden realen Ort oder Objekt verfügbar sind. Obwohl die Modellierung der virtuellen Informationen nur einmalig vorgenommen wird, kann es durchaus notwendig sein, zu einem späteren Zeitpunkt die virtuellen Informationen an Änderungen der realen Umgebung anzupassen oder Fehler zu korrigieren.

	Einsatzort	pro Anwen- dung	pro Benutzer	Eigenständiges Werkzeug (E) / Runtime Funktionalität (R)
Vorbereitung des Tracking- systems	einmalig	/	/	E
Kalibrierung von See- through- Displays	/	/	einmalige Vollkalibrie- rung, Korrek- tur bei jedem Aufsetzen	R
Modellierung der Informati- onen	/	einmalig	/	E

Tabelle 1: Häufigkeit der Engineering-Aufgaben

6.2 Template basierte Erstellung von Serviceanweisungen

Das hier beschriebene Verfahren des Template basierten Erstellens von Serviceanweisungen für AR Anwendungen erleichtert die Erstellung animierter dreidimensionaler Arbeitsbeschreibungen und zeigt einen Weg auf, den damit verbundenen Aufwand erheblich zu reduzieren. Obwohl das Verfahren spe-

ziell in Hinblick auf die Erstellung von Inhalten für AR Anwendungen entwickelt wurde, ist es ebenso für die effiziente Generierung von VR Anleitungen oder vorberechneten Videos geeignet.

Ein entscheidender Faktor für den Nutzen von Augmented Reality ist die Qualität der Inhalte, die dem Benutzer präsentiert werden. Eine klar verständliche Beschreibung der Arbeitsanweisungen vereinfacht dem Benutzer die für ihn relevanten Informationen schneller aufzunehmen. Technische Anleitungen liegen oftmals in Form von Illustrationen vor, die Schritt für Schritt durch einen Arbeitsablauf führen. Komplexe Arbeitsabläufe werden in kleine, übersichtliche Arbeitsschritte zerlegt. Der räumliche Kontext der Tätigkeit muss über die zusätzliche Darstellung für die eigentliche Tätigkeit irrelevanter Umgebungsobjekte erfolgen. Eine zu lösende Schraube kann so über prägnante, benachbarte Objekte lokalisiert werden. In AR ist dies nicht notwendig, da die Informationen per se im korrekten räumlichen Kontext dargestellt werden.

Betrachtet man exemplarisch den Service von Automobilen, so sind für einen einzelnen Hersteller mehrere Modellreihen und innerhalb dieser wiederum diverse Motor- und Ausstattungsvarianten zu beachten. Auch, wenn in verschiedenen Baureihen viele Komponenten und Baugruppen identisch sind, ist der daraus resultierende Dokumentations-Umfang immens. Zusätzliche Komplexität entsteht durch Änderungen während der Laufzeit eines Modellzyklus. Verschärft wurde das Dokumentationsproblem durch die Verkürzung von Modellzyklen und die immense Zunahme der Fahrzeugkomplexität. Besonders die Verkürzung der Modellzyklen führt logischerweise dazu, dass die Entwicklungszeit und somit auch die Zeit, die für die Erstellung der Dokumentation eines Fahrzeugs zur Verfügung steht, kontinuierlich abnehmen. Wobei durch die steigende Komplexität der Umfang der zu erstellenden Dokumentation zunimmt. Zusätzlich operieren Automobilhersteller international, so dass eine Übersetzung der Dokumentation in die verschiedenen Sprachen notwendig ist.

Den Prozessen und Werkzeugen zur Erstellung und Aufbereitung der Dokumentations-Inhalte kommt somit zunehmend eine besondere Bedeutung zu. Augmented Reality basierte Service Anweisungen in Kombination mit effektiven Autorenwerkzeugen können als ein Weg aus dem oben beschriebenen Dilemma betrachtet werden. Die hier vorgestellte Template basierte Erstellung von Service-Anweisungen stellt einen effektiven Weg der Beschreibung und Modellierung komplexer Arbeitsvorgänge anhand atomarer, generischer Aktionen dar.

6.2.1 Konzept

Eine Servicetätigkeit an einer technischen Anlage lässt sich als eine Aneinanderreihung von vielen einzelnen Arbeitsschritten darstellen. Der Ausbau eines Bauteils lässt sich meist in das Lösen von Befestigungen, wie Schrauben, und das eigentliche Entfernen des Bauteils selbst zerlegen. Zusätzlich sind vorher oftmals weitere Schritte durchzuführen, wie das Unterbrechen der Stromversorgung oder die Demontage anderer Bauteile, die den Zugang behindern. Alle diese Schritte lassen sich wiederum in einzelne, nicht mehr teilbare Aktionen, wie das Lösen einer einzelnen Schraube aufteilen. Die Hintereinanderreihung dieser atomaren Aktionen in der richtigen Reihenfolge ergibt wiederum den gesamten Arbeitsschritt. Die Arbeitsschritte, hintereinander ausgeführt, bilden die komplette Servicetätigkeit. Das Ergebnis dieser Zerlegung von Service-Aufgaben ist eine dreistufige Hierarchie (siehe Abbildung 83).

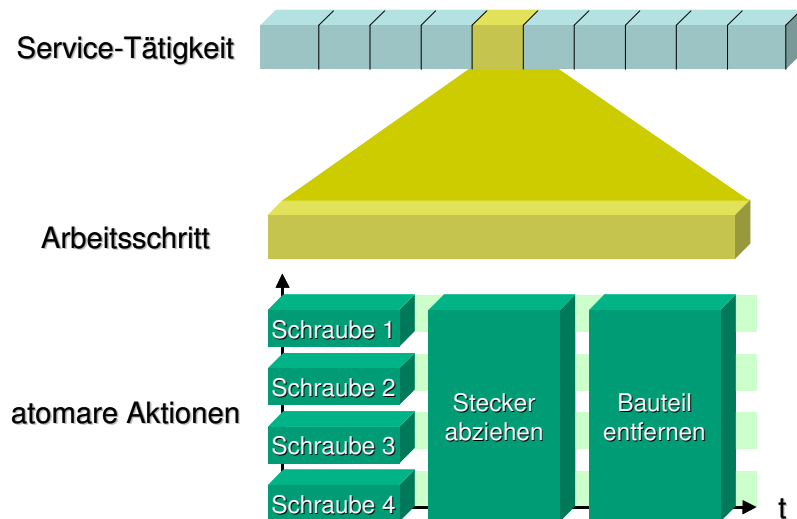


Abbildung 83: Hierarchischer Aufbau einer Service-Tätigkeit

Eine Analyse existierender Service-Anleitungen im Automobilbereich ergab, dass die Zerlegung zu einer relativ kleinen Menge atomarer Aktionen führte. Allerdings werden diese Aktionen auf eine große Menge unterschiedlicher Bauteile angewendet. An jeder Aktion ist genau ein Bauteil oder Normteil, wie eine Schraube eines bestimmten Typs, beteiligt. Zusätzlich wird oftmals noch ein Werkzeug benötigt. Im Fall des LöSENS einer Schraube ist dies ein Schraubenzieher, -schlüssel oder eine Ratsche.

Eine weitere Beobachtung ist, dass für fast alle Arbeitsschritte auch eine Umkehrung existiert. Teile die erst demontiert werden, müssen später auch wieder montiert werden. Dabei genügt es aber nicht, die einzelnen Aktionen in umgekehrter Reihenfolge auszuführen, vielmehr gibt es auch für die einzelnen Aktionen ein Inverses. Ein Arbeitsschritt wird also durch die umgekehrte Aneinanderreihung seiner inversen Aktionen umgekehrt.

Die resultierende Zerlegung in generische Aktionen, die auf spezifische Objekte unter Zuhilfenahme von Werkzeugen angewendet werden, entspricht ebenso der Begriffswelt des Servicetechnikers, der die Arbeiten durchführt, wie dem Autor, der die Anleitung erstellt. Hierbei handelt es sich in der Regel um technische Redakteure, die Experte auf dem jeweiligen technischen Gebiet sind und weniger Experten für 3D-Computergrafik. Ein Werkzeug, das ihnen effizient bei der Erstellung solcher Beschreibungen hilft, sollte somit auch den Begriffen ihrer Erfahrungswelt entsprechen. Auf diesen Sachverhalt setzt das Template Konzept auf, dass die Modellierung der Arbeitsanweisungen auf hoher Abstraktionsebene erlaubt.

Das Template Konzept selbst ist angelehnt an das gleichnamige Konzept aus der Programmiersprache C++. Templates bieten hier die direkte Unterstützung für generisches Programmieren. Ein Template ist nur von den Eigenschaften seiner Parametertypen, die es tatsächlich benutzt, abhängig [STROU98].

Jede Aktion die im Rahmen der Serviceanleitung dokumentiert ist, wird durch eine kleine 3D Animation dargestellt. Das Template stellt hierfür die Beschreibung bereit, wie diese Animation aus den Inputdaten gebildet wird.

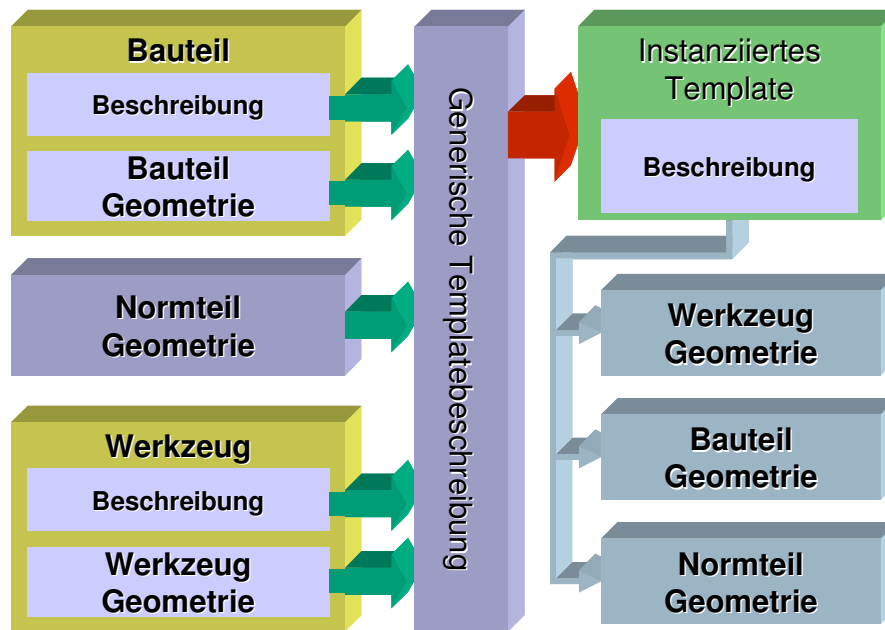


Abbildung 84: Exemplarischer Aufbau eines Templates

Zur Erzeugung einer Aktion werden die generische Template Beschreibung und die entsprechenden Inputobjekte benötigt (Abbildung 84). Beim Lösen einer Schraube wären dies beispielsweise die entsprechende Schraube (Normteil) und ein Werkzeug wie Ratsche oder Schraubenzieher. Das Template platziert nun das Werkzeug korrekt über der Schraube und wendet die Animation, die zu diesem Werkzeug vorliegt auf die Schraube an. Bei der Montage oder dem Entfernen eines Bauteils werden als Input das Bauteil selbst und der entsprechende Pfad entlang dem das Bauteil bewegt werden soll benötigt. Zusätzlich ist noch die Angabe erforderlich, in welcher Richtung eine Aktion durchgeführt werden soll. Ein Template realisiert somit die Aktion und die dazu inverse Aktion.

Zur Realisierung der Beschreibung eines Arbeitsschritts sind neben den generischen Template Beschreibungen ein hierarchisch aufgebautes Geometriemodell der Maschine sowie die Geometrien der Werkzeuge und Normteile notwendig. Voraussetzung ist weiterhin, dass alle Bauteile und die Positionen der Normteile in einem gemeinsamen Koordinatensystem vorliegen. Im Automobilbau liegen solche Modelle schon heute bei der Konstruktion eines neuen Fahrzeugtypen vor, während sie in vielen anderen Bereichen erst noch erstellt werden müssen.

Der Redakteur muss sich keine Gedanken machen, wie der Schritt am Ende dargestellt werden soll, er wählt lediglich die Aktion und die beteiligten Objekte aus. Das Autorensystem erstellt daraus automatisch die passende Darstellung und Animation. So ist auch gewährleistet, dass in einer Anleitung durchgängig die gleiche Formensprache verwendet wird und ein maximaler Wiedererkennungswert von Aktionen gewährleistet werden kann.

Die Realisierung in Form kleiner 3D Animationen ermöglicht auch weitestgehend den Verzicht auf eine sprachliche Beschreibung, so dass eine mit diesem Konzept erstellte Anleitung per se internationalisiert ist. Lediglich zusätzliche Angaben, die nicht oder nur schwer grafisch darstellbar sind, müssen als Text eingeblendet werden. Als Beispiel sei hier ein bestimmtes Drehmoment, mit dem eine Schraube angezogen werden muss, genannt.

Ein weiterer wichtiger Aspekt ist der hohe Grad an Wiederverwendbarkeit. Dies gilt im speziellen für die generischen Templates, aber auch für die damit erstellten Arbeitsschritte und Aktionen, so dass Arbeitsschritte, die auch in anderen Servicetätigkeiten vorkommen bei deren Beschreibung einfach erneut referenziert werden können.

6.2.2 Realisierung

Umgesetzt wurde das Konzept der Template basierten Erstellung von Service Anweisungen basierend auf VRML Knoten [VRML97]. Vorteil der Realisierung auf Basis von VRML-Knoten ist die schnelle Erstellung neuer Knoten oder die Anpassung bestehender Knoten ohne ein erneutes Übersetzen von Programmcode. Darüber hinaus lässt sich das Ergebnis in jedem Standard VRML Browser darstellen und es existiert eine ausführliche Dokumentation der VRML Mechanismen.

VRML unterstützt die Definition eigener, angepasster Knotentypen über das EXTERNPROTO Konzept. Ein EXTERNPROTO enthält die Beschreibung des externen Interface und die Implementierung, die wiederum durch VRML-Knoten realisiert wird. Zu diesen Knoten gehören auch Scriptknoten, in denen das dynamische Verhalten realisiert ist. Das Interface des selbst definierten Knotentyps wird durch dessen Felder repräsentiert. Diese Felder ermöglichen den Zugriff auf den sichtbaren Zustand des Knotens und nehmen die Objekte, die an einer Aktion beteiligt sind auf. Für die Erstellung einer Aktion sucht der Redakteur das passende Template aus und wählt anschließend das Bauteil oder Normteil und das notwendige Werkzeug aus. Das Autorensystem instanziiert daraus die komplette Aktion. Bei den Aktionen ist hier zwischen solchen zu unterscheiden, die automatisch instanziiert werden und solchen, die weitere Interaktionen des Redakteurs verlangen. Beim Entfernen eines Bauteils lässt sich beispielsweise die Bewegungsbahn vorgeben, entlang der es transformiert werden soll.

Da ein Arbeitsschritt in der Regel aus mehreren Aktionen besteht, die in einer bestimmten Reihenfolge auszuführen sind, wird auf dieser Ebene eine Möglichkeit benötigt, die zeitliche Abfolge der einzelnen Aktionen zu steuern. Aktionen können hintereinander, parallel oder zeitlich überlappend ausgeführt werden. Jede Aktion besitzt eine lokale Zeitsteuerung, die wiederum von der zeitlichen Steuerung auf Ebene des übergeordneten Arbeitsschritts gesteuert wird. Da VRML zur Zeitsteuerung normierte Intervalle ($t \in [0;1]$) verwendet, ist es erforderlich, die globale Zeit des Arbeitsschritts in die entsprechenden Intervalle der Aktionen zu transformieren. Realisiert ist dies wiederum in speziellen PROTO Knoten. Neben der zeitlichen Steuerung der Animationen in den Aktionen wird über dieses Konzept auch die Sichtbarkeit von Bauteilen, Normteilen und Werkzeugen gesteuert, so dass immer nur die für eine Aktion notwendigen Objekte sichtbar sind.

Ist ein Arbeitsschritt im Autorensystem fertig gestellt, wird er in Form einer XML-Beschreibung gespeichert, die Verweise auf alle verwendeten Templates und deren Parametrierung enthält. Wird anschließend eine Template Beschreibung geändert, wirkt sich diese Änderung auf alle Arbeitsschritte aus, die dieses Template referenzieren, ohne dass diese editiert werden müssen.

Auf Seiten des Runtime Systems wird die Beschreibung der Arbeitsschritte eingelesen und die darin enthaltenen Aktionen über die jeweiligen Templates instanziiert. So kann sichergestellt werden, dass Aktionen, die aus dem gleichen Template erzeugt wurden, auch auf der Runtime Seite gleich dargestellt werden.

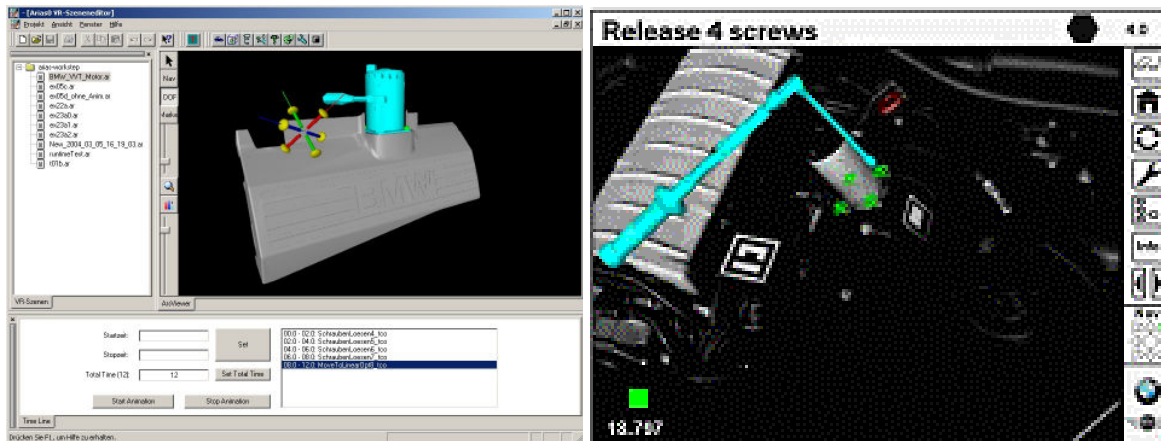


Abbildung 85: Darstellung in Autorentool und Laufzeitsystem

Ein wesentlicher Unterschied in der Darstellung zwischen Autorentool und Laufzeitsystem besteht darin, dass aufgrund der rein virtuellen Arbeitsumgebung im Autorentool die Darstellung zusätzlicher Geometrien notwendig ist (siehe Abbildung 85). Diese zusätzlichen Geometrien geben dem Redakteur den Kontext der aktuellen Aktion und erleichtern ihm so die Orientierung in der Szene. Am Laufzeitsystem werden diese zusätzlichen Objekte dagegen nicht dargestellt, da hier die entsprechenden realen Objekte vorhanden sind.

Um die Arbeitsschritte lagerichtig in das Blickfeld des Servicetechnikers vor dem realen Auto einblenden zu können ist es notwendig, dass die Tracking-Informationen des AR-Systems dem Koordinatensystem entsprechen, das der Erstellung der Arbeitsschritte im Autorentool zugrunde lag. Im Automobilservice bietet sich hier das Fahrzeugkoordinatensystem an, in dem sich bereits die Beschreibungen aller Bau- und Normteile befinden. Das verwendete Tracking-System wird beim Einrichten auf dieses Koordinatensystem kalibriert. Im Falle von markerbasiertem Videotracking bedeutet dies, dass die Marker relativ zum Fahrzeugkoordinatensystem vermessen werden müssen, dass auch dem virtuellen Fahrzeugmodell zugrunde liegt.

6.2.3 Erweitertes Template Konzept

Vorteil der Ablage realisierter Templates als VRML-Beschreibungen ist, dass diese auch auf Standard-VRML-Viewern angezeigt werden können. Allerdings ist bei dieser Form der Speicherung von Arbeitsanweisungen die eigentliche Beschreibung eines Arbeitsschritts fest mit der Darstellung verbunden. Das Ändern der Darstellung erfordert das Ändern des verwendeten Templates und das erneute Durchführen des Authoring-Schritts.

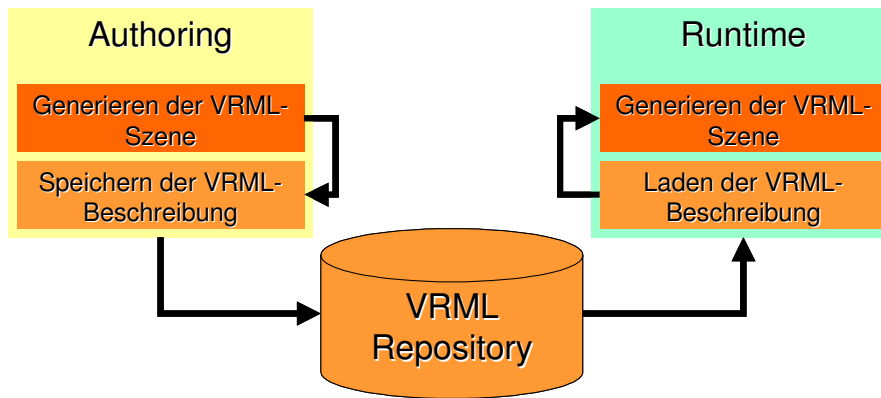


Abbildung 86: Erzeugen der Arbeitsanweisungen als VRML-Beschreibung

Eine dynamische Anpassung der Darstellung an unterschiedliche Benutzer- oder Hardwareprofile ist nach der Erzeugung der Beschreibung im VRML-Format nicht mehr möglich. An dieser Stelle greift das erweiterte Template-Konzept, bei dem die Kommunikation zwischen Authoring- und Runtime-System nicht mehr über VRML-Beschreibungen vorgenommen wird. Stattdessen wird eine Beschreibung übergeben, aus der zur Laufzeit die Darstellung generiert wird. Die Komponente zur Erzeugung der graphischen Präsentation wird somit vom Authoring-System auf die Runtime-Seite verlagert. Die Komponente zur Erzeugung von VRML-Szenen und deren Darstellung dient auf der Seite des Authoring-Systems lediglich noch als Preview-Funktionalität.

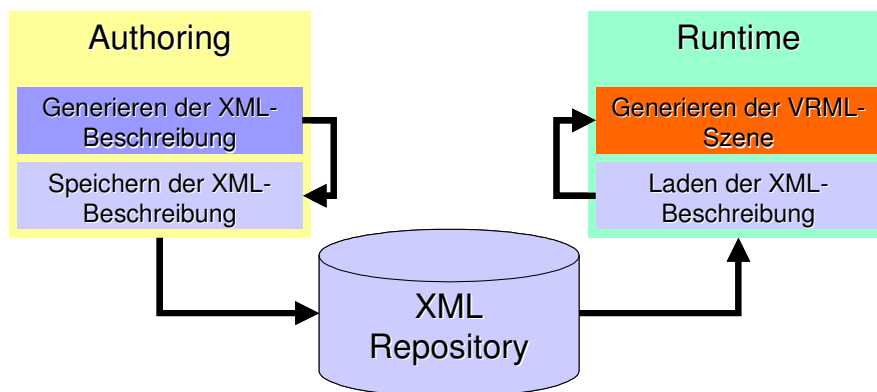


Abbildung 87: Erzeugen der Arbeitsanweisung als abstrakte XML-Beschreibung

Die Beschreibung des Arbeitsschritts ist von dessen Darstellung entkoppelt. Notwendig für die Entkopplung ist die Definition einer geeigneten Beschreibungssprache, die es ermöglicht daraus wieder eine adäquate Darstellung zu generieren. Von Vorteil hierbei ist wiederum die relativ kleine Menge atomarer Aktionen, die zur Komposition der häufigsten Arbeitsschritte benötigt werden. Das Vokabular der Beschreibungssprache kann somit entsprechend kompakt gehalten werden. Es besteht aus den verfügbaren Aktionen und deren Parametrierung.

Auf der Seite des Runtime Systems werden die abstrakten XML-Beschreibungen der Aktionen interpretiert und gemäß des dort aktiven Hardware- und Benutzerprofils eine angepasste Präsentation erzeugt. So kann aus der gleichen XML-Beschreibung, die das Lösen einer Schraube beschreibt, entwe-

der eine abstrakte Darstellung durch rotierende Pfeile erzeugt werden oder eine eher naturalistische Darstellung mit animierten Schrauben- und Werkzeuggeometrien.

Erkauft wir der Gewinn an Flexibilität und Anpassbarkeit der Präsentation mit einem Verlust an Kontrolle auf Seiten des Redakteurs, der in der Regel die Dokumentation eines Arbeitsschritts auf eine bestimmte Darstellungsform hinoptimiert.

6.2.4 Online Authoring

Im Gegensatz zum oben beschriebenen Offline Prozess, bei dem der Anwender auf fertig generierte Daten zugreift, wird im folgenden ein alternativer Prozess beschrieben, der überall dort nützlich ist, wo keine oder nur unzureichende Information für AR Präsentationen erzeugt wurden. Erreicht ein Techniker bei einer Service-Aufgabe eine Situation, in der ihm die verfügbare Dokumentation nicht mehr weiterhilft, ist er auf die Hilfe eines Experten angewiesen, der allerdings meist nicht vor Ort verfügbar ist. Was zur Konsequenz hat, dass der Experte über ein verfügbares Kommunikationsmedium, meist Telefon, zugeschaltet wird. Der Techniker muss in diesem Fall die Situation komplett verbal schildern und der Experte seine Diagnose aufgrund dieser Beschreibung stellen, was vor allem bei komplexen Anlagen fehleranfällig ist und leicht zu Missverständnissen führt.

Zu einer Verbesserung der Situation führt die Übertragung von Standbildern und Videodaten der Situation vor Ort, so dass der Experte sich selbst basierend auf diesen Daten ein eigenes Bild machen kann und sich mit seinen Anweisungen auf das Gesehene beziehen kann. Geht man davon aus, dass der Techniker vor Ort eine Kamera am Kopf, an einem HMD trägt, so kann der Experte die Sicht des Technikers teilen. Werden zusätzlich auch die Tracking-Informationen vom System des Technikers übertragen, kann der Experte in einem gewissen Maße auch die erweiterte Realität teilen, in der sich der Techniker befindet. Er hat nun die Möglichkeit auch in dieser geteilten erweiterten Realität zu agieren, seine Hilfestellungen direkt in 3D in Form virtueller Annotationen in der Umgebung des Technikers anzubringen. Zur Effizienten Unterstützung des Experten bei dieser Tätigkeit ist es nun notwendig, ihm effiziente Werkzeuge an die Hand zu geben, diese Annotationen zu erstellen. Nahe liegend sind hierbei die Verwendung des oben beschriebenen Autorenwerkzeugs und seine Erweiterung um die speziellen Funktionalitäten, die für das Online-Authoring benötigt werden.

Während bei der klassischen Variante des Offline-Authorings in einer rein virtuellen Umgebung gearbeitet wird, findet die Generierung von Objekten im Online-Authoring-Prozess in einer AR Umgebung statt. Damit der Remote-Experte die erweiterte Realität des Technikers vor Ort teilen kann, ist eine Übertragung sowohl der Bild- als auch der Tracking-Daten vom Rechner des Technikers vor Ort zum Remote-Experten notwendig. Die zusätzliche Übertragung der Tracking-Informationen ermöglicht auch die Verwendung anderer, als reine bildbasierte Tracking-Verfahren, wie die Kombination von bildbasiertem Tracking und Inertial-Sensoren. Auch die Komprimierung der Bilddaten kann so die Tracking-Ergebnisse nicht mehr negativ beeinflussen.

Der Remote-Experte bewegt sich virtuell synchron mit dem Techniker durch dessen Umgebung und besitzt durch seinen Desktop-Arbeitsplatz die Möglichkeit einfach und präzise virtuelle Objekte in der realen Umgebung anzubringen. Diese Objekte werden nach ihrer Erzeugung und Parametrierung dann vom Remote-Experten zum Techniker vor Ort übertragen.

6.3 Zusammenfassung

Die Engineeringaufgaben in einem mobilen AR System lassen sich in die folgenden drei Kategorien aufteilen:

- Einrichten und Kalibrieren von Trackingsystemen
- Kalibrierung von See-Through-Displays
- Modellierung des virtuellen Informationsraums

Die Konzentration des hier vorgestellten Konzepts liegt auf der Modellierung des virtuellen Informationsraums mit dem besonderen Fokus auf der Erstellung der Service-Dokumentation für komplexe technische Güter. Dabei werden komplexe Arbeitsanweisungen in einfache Aktionen heruntergebrochen, für die dann semi-automatisch die entsprechenden Animationen erstellt werden können. Als Bestandteile einer solchen Animation werden einzig vorgefertigte Aktions-Templates, die universell wiederverwendbar und parametrierbar sind, und die entsprechenden Werkstücks- und Werkzeugs-Geometrien benötigt. Als zentrales Element des Konzepts haben die Aktions-Templates dem Verfahren auch seinen Namen, Template basiertes Autoring gegeben.

Das Ergebnis eines solchen Authoringschrittes ist eine VRML-Animation, die neben dem AR Viewer auch in diversen anderen Umgebungen abgespielt werden kann. Die zusätzliche Möglichkeit, die Animation auch als XML-Beschreibung abzulegen, ermöglicht die Dokumentation zur Laufzeit an die Vorlieben und Bedürfnisse eines Anwenders anzupassen oder eine für das verwendete Display passende Form der Darstellung zu wählen.

Als zusätzlicher Nutzen der vorgestellten Techniken zur Erstellung von Service Anweisungen, lassen sich diese auch für den Remote-Service nutzen, bei dem ein entfernter Experte die Anweisungen erstellt, die dem Techniker vor Ort in das Sichtfeld eingeblendet werden.

Das vorgestellte Konzept des Template basierten Authorings und seiner Erweiterungen lässt sich universell einsetzen und hat auch bereits in der Praxis das Potential zur Einsparung von Zeitaufwänden bei der Erstellung von animierten Anleitungen zu komplexen technischen Gütern bewiesen.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Den Schwerpunkt der vorliegenden Arbeit bilden die Konzeption und Realisierung einer tragfähigen Softwarearchitektur für die Erstellung mobiler Augmented Reality Anwendungen sowie die Erarbeitung intuitiver Interaktionstechniken zur Unterstützung der Arbeit mit mobile AR Anwendungen. Neben dem Laufzeitsystem wurde dabei die komplette Prozesskette, also auch die Schritte zur Vorbereitung einer AR Anwendung betrachtet.

Grundlage für die hier vorgestellte Systemarchitektur bildet eine Umfassende Erhebung der Anforderungen stellvertretend in vier repräsentativen Anwendungsfeldern für mobile AR Anwendungen. Das Hauptaugenmerk bei dieser Betrachtung liegt dabei auf Funktionalitäten, die einen Benutzer bei der Durchführung einer realen Tätigkeit unterstützen, den AR basierten Assistenzfunktionalitäten.

Die Analyse der gewonnenen Anforderungen ergab dabei neben speziellen Anforderungen an die Basisfunktionalitäten, wie Tracking, Präsentation und Interaktion auch die Notwendigkeit zusätzlicher Funktionalitäten. Basierend auf den Analyseergebnissen entstand das Augmented Reality Basissystem AR-Browser, dessen komponentenbasierte Architektur und die darin realisierten Komponenten besonders in Hinsicht auf die Verwendung in mobilen Anwendungen ausgelegt sind.

Die Applikationsschnittstelle des AR-Browsers ist speziell auf die Integration in webbasierte Applikationen ausgelegt. Der AR-Browser kann in bestehende HTML Seiten eingebettet und über das Scriptinterface administriert werden. Die benötigten Applikationsdaten kann der AR-Browser dabei selbstständig über das Http Protokoll beziehen.

Die Organisation der Daten erfolgt unabhängig von der Struktur des reinen Render-Szenengraphen. So ist es möglich unabhängig von der Qualität und Granularität der 3D Modellierung den virtuellen Objekten in der Umgebung des Benutzers einen Informationsraum aufzubauen, der sich an der Semantik der realen Umgebung orientiert. Zusätzlich erlaubt dieses erweiterte Weltmodell auch das verknüpfen von Informationsobjekten mit realen Objekten, für die kein entsprechendes Geometriemodell existiert.

Basierend auf diesen zusätzlichen Strukturen ist auch eine weitere Funktionalität, die Informationsfilterung realisiert. Diese Filtermechanismen sind als Erweiterung der klassischen Level of Detail Techniken der Computergrafik anzusehen, wobei neben dem Abstand zu einem Informationsobjekt auch weitere Attribute eines Objektes berücksichtigt und mit dem aktuellen Zustand des Anwenders verrechnet werden. Das Ergebnis ist ein Vektor dessen Einzelelemente gewichtet die Sichtbarkeit steuern.

Dient die Informationsfilterung der Steuerung der Menge sichtbarer Datenobjekte, sind für mobile AR Anwendungen einige zusätzliche Aspekte der Darstellung virtueller Objekte von Bedeutung, die in

klassischen VR Systemen, wie dem als Basis dienenden Avalon nicht vorhanden sind. Dabei handelt es sich beispielsweise um die Berücksichtigung visueller Interaktionen zwischen realer und virtueller Welt, wie etwa Schatten. Außerdem gilt es für unterschiedliche Informationstypen die geeignete Darstellungsform zu wählen, so dass sie ein Anwender schnellstmöglich erfassen kann.

Ein weiterer wichtiger Aspekt der Architektur eines AR Systems ist die Anbindung von Tracking-Systemen zur Bestimmung von Position und Blickrichtung des Benutzers. In mobilen Umgebungen ist dies nur unter Verwendung von bildbasierten Tracking-Verfahren mit ausreichender Präzision möglich. Da diese aber hohe Rechenlast und Verzögerungen erzeugen ist die Verwendung zusätzlicher Tracking-Sensorik notwendig. Hierbei kommen meist Inertial-Sensoren mit hoher Updaterate zum Einsatz, so dass zumindest die Blickrichtung auch beim Ausfall des primären bildbasierten Tracking-Verfahrens bestimmt werden kann. Zur Realisierung solcher Verfahren wurde ein Tracking-Framework entwickelt, das neben maximaler Performanz vor allem die leichte Austauschbarkeit von einzelnen Komponenten, also Tracking-Verfahren und deren Konfiguration erlaubt.

Neben der Systemarchitektur besteht der zweite Schwerpunkt auf der Laufzeit-Seite des Systems auf der Bedienbarkeit eines AR Systems in einer mobilen Umgebung. Sowohl klassische Desktop Eingabegeräte als auch spezielle VR Interaktionsgeräte, die meist eine spezielle Tracking-Infrastruktur voraussetzen, sind hier nicht einzusetzen. Dieser Umstand führte zur Entwicklung eines universellen Eingabegerätes für den mobilen Einsatz in AR Umgebungen. Das Gerät kann dabei als Erweiterung des Personal Interaction Panels (PIP) der Studierstube angesehen werden, dass durch die Notwendigkeit von 6D Tracking nur im Blickfeld des Benutzers eingesetzt werden kann. Das hier vorgestellte Eingabegerät kann durch speziell angepasste Interaktionstechniken und die Ausnutzung der Propriozeption des Anwenders auch außerhalb des Blickfeldes des Benutzers für typische Interaktionsaufgaben in 3D Umgebungen wie die Objektselektion verwendet werden.

In Verbindung mit dem Interaktionsgerät wurde auch ein Konzept erarbeitet und umgesetzt, dass die Integration graphischer Benutzerschnittstellen in der szenengraphbasierten Umgebung erlaubt. Spezielle Knoten im Szenengraph übernehmen dabei die Verbindungslogik während die grafische Repräsentation durch davon getrennte Geometriknoten vorgenommen wird. Die Propagierung von Events geschieht über die Mechanismen des darunter liegenden VRML Szenegraphen, über Felder und Routen. Das Konzept dieser 3D Benutzerschnittstellen erlaubt die Generierung von GUI Elementen sowohl in fester Relation zu einem Objekt als auch display-fix mit festen Bildschirmkoordinaten. Die verwendeten Interaktionsmechanismen bleiben davon unabhängig immer die gleichen. Auch die Unabhängigkeit von speziellen Eingabegeräten konnte erreicht werden.

Wie bereits eingangs angesprochen, wurde neben dem Laufzeitsystem auch die komplette Prozesskette betrachtet, die zum Einsatz von AR Techniken notwendig ist. Auch hier konnten Ergebnisse der Anforderungserhebung genutzt werden, die auch die wichtigsten Benutzergruppen, also auch System- und Anwendungsentwickler berücksichtigte. Besonderes Augenmerk wurde dabei auf das zeitaufwändige Erstellen der virtuellen Szene und hier im Speziellen von virtuellen Arbeitsanweisungen gelegt. Das hier vorgestellte Verfahren des Template basierten Authorings beruht auf der Beobachtung, dass solche Anweisungen meist aus ähnlichen Aktionen zusammengesetzt werden, die allerdings unterschiedlich parametrisiert werden und auf verschiedene Werkstücke anzuwenden sind. Das entsprechende Autorenwerkzeug erlaubt danach das Anwenden atomarer Aktionen wie „Schraube lösen“ auf ein

Bauteil. Der technische Redakteur wählt die Aktion, wendet sie auf das Bauteil an und selektiert noch das notwendige Werkzeug. Das Programm erzeugt ihm daraus die fertige Animation zur Illustration der Tätigkeit. Der Effekt ist eine enorme Zeiteinsparung bei der Erstellung solcher Anweisungen für VR und AR Umgebungen. Eine Erweiterung des Verfahrens fügt dabei noch zusätzliche Flexibilität hinzu, da die eigentliche Animation erst auf dem Laufzeitsystem erzeugt wird. Die eigentliche Darstellung kann somit abhängig von Hardware- und Benutzerprofilen erzeugt werden.

7.2 Ausblick auf weiterführende Arbeiten

Die vorgestellten Konzepte wurden allgemein für die Realisierung mobiler AR Basissysteme erstellt eine Realisierung liegt bisher aber lediglich für Rechner mit Desktop Betriebssystemen vor. Eine Umsetzung und anschließende Evaluierung auf mobilen Rechner der PDA Klasse steht noch aus. Auch die Erschließung zusätzlicher Anwendungsgebiete wird dazu führen, die bestehende Architektur zu reflektieren und Anpassungen vorzunehmen.

Das Konzept der Informationsobjekte hat sich für verschiedene Techniken, wie zum Beispiel die Informationsfilterung, als sehr hilfreich erwiesen. Es werden allerdings auch hier Werkzeuge benötigt, die möglichst effizient beim Aufbau der Strukturen, dem Füllen der Attribute und bei der Herstellung der Verknüpfung mit dem Szenengraphen helfen. Generell wird den Engineering-Tools wesentlich mehr Aufmerksamkeit geschenkt werden müssen, wenn sich der Einsatz von AR Technologien auf breiter Front etablieren soll.

Neben der Erstellung von Inhalten ist auch die Bedienung der Systeme weiterhin kritisch zu betrachten. Das vorgestellte Eingabegerät bietet zusammen mit den damit verbundene intuitiven Eingabetechiken zwar weitgehende Funktionalitäten, ist aber auch aufgrund des prototypischen Aufbaus noch relativ unhandlich.

Ein wachsender Markt für AR Anwendungen und Geräte, wie er bei dem Potential der Technik und der zumindest softwareseitigen Entwicklung wird sich auch das generelle Interesse an geeigneter Hardware vergrößern. Auch hier kann man ähnlich dem 3D Grafikboom bei Desktoprechnern auf die Entertainment-Industrie hoffen.

Literaturverzeichnis

- [AZUMA97] Azuma, R.: "A Survey of Augmented Reality," Presence: Teleoperators and Virtual Environments. vol. 6, no. 4, pp. 355-385, August 1997
- [AZUMA99] Azuma, R., Hoff, B., Neely III, H., Sarfaty, R.: "A Motion-Stabilized Outdoor Augmented Reality System". Proceedings of IEEE VR 99, pp. 252-259 Houston, 1999
- [BAJU92] Bajura, M., Fuchs, H., Ohbuchi, R.: "Merging Virtual Reality with the Real World: Seeing Ultrasound Imagery Within the Patient.". Proceedings of SIGGRAPH 92, Chicago, 1992
- [BARA02] Baratoff, G., Neubeck, A., Regenbrecht, H.: „Interactive Multi-Marker Calibration for Augmented Reality Applications“, Proceedings of ISMAR 02, Darmstadt, 2002.
- [BAUER01] Bauer, M., Brügge, B., Klinker, G., MacWilliams, A., Reicher, T., Riss, S., Sandor, C., Wagner, M.: "Design of a Component-Based Augmented Reality Framework", Proceedings of ISAR 01, New York, 2001.
- [BAUER02] Bauer, M., Brügge, B., Klinker, G., MacWilliams, A., Reicher, T., Sandor, C., Wagner, C.: "An Architecture Concept for Ubiquitous Computing Aware Wearable Computers", International Workshop on Smart Appliances and Wearable Computing (IWSAWC 2002), Wien, 2002.
- [BEDER95] Bederson, B.: "Audio Augmented Reality: a prototype automated tour guide". Conference companion on Human factors in computing systems, 1995, pp 210.
- [BEHR02] Behr, J., Eschler, P., Fröhlich, T., Knöpfle, C., Lutz, B., Müller, S., Roth, M.: "Cybernarium Days 2002 – A Public Experience of Virtual and Augmented Worlds", Proceedings of the First International Symposium on Cyber Worlds, (CW 02), pp. 553-560, Los Alamitos, 2002
- [BEHR04] Behr, J., Dähne, P., Roth, M.: "Utilizing X3D for immersive environments", Web3D '04: Proceedings of the ninth international conference on 3D Web technology, 71-78, Monterey, 2004
- [BEHRING02] Behringer, R., Park, J., Sundareswaran, V.: "Model-Based Visual Tracking for Outdoor Augmented Reality Applications", Proceedings of ISMAR 02, Darmstadt, 2002
- [BELL01] Bell, B., Feiner, S., Höllerer, T.: "View Management for Virtual and Augmented Reality". Proceedings of UIST 2001 (ACM Symp. on User Interface Software and Technology), Orlando, 2001, pp. 101-110

- [BENF93] Benford S., Fahlén, L.: "A Spatial Model of Interaction in Large Virtual Environments. In Proceedings of ECSCW '93, Mailand, 1993.
- [BIER93] Bier, E., Stone, M., Pier, K., Buxton, W., DeRose, T.: "Toolglass and Magic Lenses: The See-Through Interface", Proceedings of 1993 ACM SIGGRAPH Conference, 73-80
- [BIERB01] Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., Cruz-Neira, C.: „VR Jugler – A Virtual Platform for Virtual Reality Application Development”, In IEEE Virtual Reality 2001, pp 89-96, Yokohama, 2001
- [BILL98] Billingham, M., Bowskill, J., Dyer, N., Morphet, J.: "An Evaluation of Wearable Information Spaces.", In Proceedings of IEEE VRAIS 98, Atlanta, Georgia, 1998, IEEE Computer Society Press
- [BIMB03] Bimber, O., Encarnação, L.M., and Schmalstieg, D. "The Virtual Showcase as a new Platform for Augmented Reality Digital Storytelling". Proceedings of IPT/EGVE 2003 Workshop, 2003.
- [BLON96] Blonde, L., Buck, M., Galli, R., Niem, W., Paker, Y., Schmidt, W., Thomas, G.: "A Virtual Studio for Live Broadcasting: The Mona Lisa Project", IEEE MultiMedia, Summer 1996, (Vol. 3, No. 2), pp. 18-29, 1996
- [BÖHM96] Böhm, K.: „Ein generisches 3D-User Interface Toolkit mit Verfahren zur Gebärdenerkennung“, Dissertation, Technische Hochschule Darmstadt, 1996
- [BOW01] Bowman, D. A., Wingrave, C. A.: „Design and Evaluation of Menu Systems for Immersive Virtual Environments“, in Proceedings of IEEE-VR 2001, pp. 149-156, Yokohama, 2001
- [BRED99] Brederson, J.: "The I3Stick: An Inexpensive, Immersive, Interaction Device.". *Technical Report*, UUCS-99-016. University of Utah. November 1999.
- [BROWN95] Brown, P. J., "The electronic post-it note: a metaphor for mobile computing applications", IEEE Colloquium on Mobile Computing and its Applications, 1995.
- [CARD83] Card, S. K., Moran, T. P., Newell, A.: „The psychology of human-computer interaction“, Hillsdale, NJ, Lawrence Erlbaum Associates, 1983
- [CAU92] Caudell, T. P., Mizell, D. W.: "Augmented Reality: An Application Of Heads-Up Display Technology to Manual Manufacturing Processes", Proceedings of Hawaii International Conference on System Sciences, 1992, 659-669.
- [CRUZ92] Cruz-Neira, C., Sandin, D., Defanti, T., Kentyon, R., Hart, J.: "The CAVE: Audio Visual Experience Automatic Virtual Environment". Communications of the ACM, 1992, Vol. 35 (6), pp. 65.
- [FEIN93] Feiner, S., MacIntyre, B. and Seligmann, D.: "Knowledge-based Augmented Reality.", Communications of the ACM Vol. 36 (7) (July 1993), 52-62.

- [FEIN97] Feiner, S., MacIntyre, B., Höllerer, T., Webster, A.: "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment". Proceedings of First International Symposium on Wearable Computers (ISWC '97)Cambridge, 74-81.
- [FELGER95] Felger, W: "Innovative Interaktionstechniken in der Visualisierung", Dissertation Technische Hochschule Darmstadt, 1995
- [FER01] Ferrari, V., Tuytelaars, T., Van Gool, L.: "Markerless Augmented Reality with a Real-Time Affine Region Tracker," Proc. IEEE and ACM Int'l Symp. Augmented Reality, vol. I, IEEE Computer Soc. Press, Los Alamitos, 2001, pp. 87-96.
- [FOL90] Foley, J., van Dam, A., Feiner, S., Hughes, H.: „Comuter Graphics - Principles and Practice“, Addison Wesley, 1990.
- [FRÖHL00] Fröhlich, T.; Roth, M.: "Integration of Multidimensional Interaction Devices in Real-Time Computer Graphics Applications", Computer Graphics Forum (Eurographics'00 Proc.), 15(3): 267-273, 2000
- [GABB02] Gabbard, J., Swan, J., Hix, D., Lanzagorta, M., Livingston, M., Brown, D., Julier, S.: "Usability Engineering: Domain Analysis Activities for Augmented Reality Systems". Proceedings SPIE Vol. 4660, p. 445-457, Stereoscopic Displays and Virtual Reality Systems IX, Photonics West 2002, Electronic Imaging conference, San Jose, January 19-25, 2002.
- [GEIG03] Geiger, C., Oppermann, L., Reimann, C: „3D-Registered Interaction-Surfaces in Augmented Reality Space“, In the Second IEEE International Augmented Reality Toolkit Workshop, Tokyo, 2003.
- [GENC02] Genc, Y., Riedel, S., Souvannavong, F., Akmlar, C., Navab, N.: "Marker-less Tracking for AR: A Learning-Based Approach", Proceedings of ISMAR02, Darmstadt, 2002
- [GUEV03] Güven, S, Feiner, S.: "Authoring 3D Hypermedia for Wearable Augmented and Virtual Reality“, Proceedings of the Seventh IEEE International Symposium on Wearable Computers (ISWC'03), 2003
- [HÄF00] Häfner, U. Bues, M., Magg, R.: "Wireless interaction in cost-effective display environments", International Immersive Projection Technology Workshop, 19.-20.6.2000, Ames/Iowa
- [HALL02] Haller, M., Hartmann, W., Luckeneder, T., Zauner, J.: "Combining ARToolKit with Scene Graph Libraries", Proceedings of ARToolkit Workshop ART02, Darmstadt, 2002
- [HALL03] Haller, M., Drab, S., Hartmann, W.: „A realtime shadow approach for an Augmented Reality application using shadow volumes“, VRST03

- [HARI02] Haringer M., Regenbrecht, H.: "A Pragmatic Approach to Augmented Reality Authoring," Proceedings of ISMAR '02 (IEEE and ACM Int. Symp. on Augmented Reality), Germany, 2002, 237–245.
- [HINCK94] Hinckley, K., Pausch R., Goble, J., Kassel, N.: „A Survey of Design Issues in Spatial Input“, in Proceedings of UIST '94, 1994, pp. 213-222
- [HOELL99a] Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., Hallaway, D.: "Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System", Computers & Graphics 23(6), pp. 779-785, 1999
- [HOELL99b] Höllerer, T., Feiner, S., Pavlik, J.: „Situating Documentaries: Embedding Multimedia Presentations in the Real World“, Proceedings of ISWC 99, pp. 79-86, San Francisco, 1999
- [HOELL01] Höllerer, T., Feiner, S., Hallaway, D., Bell, B.: „User Interface Management Techniques for Collaborative Mobile Augmented Reality“, Computers and Graphics 25(5), Elsevier Science Ltd, Oct. 2001, pp. 799-810
- [IDDAN01] Iddan, G., Yahav, G.: "3D Imaging in the Studio (and Elsewhere ...)", Proceedings of SPIE 4298: Videometrics and Optical Methods for 3D Shape Measurements, 48-55, 2001.
- [ISHI97] Ishii, H., Ulmer, B.: "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms", Proceedings of CHI 97, ACM Press, pp. 234-241, 1997
- [JOHN88] Johnson, R.; Foote, B.: "Designing Reusable Classes. Journal of Object-Oriented Programming", 1(2), 22-35, 1988
- [JULI00] Julier, S., Lanzagorta, M., Baillot, Y., Rosenblum, L., Feiner, S., Höllerer, S., Sestito, S.: "Information filtering for mobile augmented reality", Proceedings of ISAR 00, pp 3–11, München, 2000.
- [KATO99] Kato, H., Billinghurst, M.: "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System", Proceedings of IWAR 99, San Francisco, 1999.
- [KATO03] Kato, H., Tachibana, K., Tanabe, M., Nakajima, T., Fukuda, Y.: "A City-Planning System based on Augmented Reality with a Tangible Interface", Proceedings of ISMAR 03, pp. 340-341, 2003
- [KLI98] Klinker, G., Stricker, D., Reiners, D.: „The Use Of reality Objects In Augmented Reality Applications“, European Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE), Freiburg, 1998.
- [KNÖ03] Knöpfle, C.: "Intuitive und immersive Interaktion für virtuelle Umgebungen am Beispiel von VR-Design Review ", Dissertation, Technische Universität Darmstadt, 2004
- [LEON96] Leonhard, U., Magee, J., Dias, P. Location Service in Mobile Computing Environments. Computer & Graphics. Special Issue on Mobile Computing. Volume 20, Nummer 5, 1996.

- [LIND99a] Lindeman, R., Sibert, J., Hahn, J.: „Towards Usable VR: An Empirical Study of User Interfaces for Immersive Virtual Environments“, in Proceedings of SIGCHI 99, pp. 64-71, 1999
- [LIND99b] Lindeman, R., Sibert, J., Hahn, J.: “Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments”, in Proceedings of IEEE VR 99, Houston, pp. 205-212, 1999
- [LIVI02] Livingston, M., Rosenblum, L., Julier, S., Brown, D., Baillot, Y., Swan II, J. E., Gabbard, J., Hix, D.: ”An Augmented Reality System for Military Operations in Urban Terrain,” Proceedings of the Interservice / Industry Training, Simulation, & Education Conference (I/ITSEC '02), Orlando, 2002
- [MAGE] Magellan 3D Controller - User's Manual Version 4.2 Logitech, 1995.
- [NEUM98] Neumann, U., Majoros, A.: “Cognitive, Performance, and Systems Issues for Augmented Reality Applications in Manufacturing and Maintenance”, in IEEE VRAIS '98, Atlanta, 1998
- [OHSH03] Ohschima, T., Kuroki, T., Yamamoto, H., Tamura, H.: “A Mixed Reality System with Visual and Tangible Interaction Capability - Application to Evaluating Automobile Interior Design -“, ART03
- [OTUG03] OpenTracker Users Guide: <http://www.studierstube.org/opentracker/>
- [PASM03] Pasman, W., Woodward, C.: “Implementation of an Augmented Reality System on a PDA”, International Symposium on Mixed And Augmented Reality, Proceedings of ISMAR03, Tokyo, 2003
- [PIEK01a] Piekarski, W., Thomas, B.: “Tinmith-evo5 – An Architecture for Supporting Mobile Augmented Reality Environments”. In 2nd Int'l Symposium on Augmented Reality, New York, 2001
- [PIEK01b] Piekarski, W., Thomas, B.: “Tinmith-Metro: New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer”, In *5th Int'l Symposium on Wearable Computers*, pp 31-38, Zürich, 2001
- [PIEK02] Piekarsky, W., Thomas, B.: “Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments“. ART02, 1st International Augmented Reality Toolkit Workshop, Darmstadt, 2002
- [PIEK03] Piekarski, W., Thomas, B.: “Interactive Augmented Reality Techniques for Construction at a Distance of 3D Geometry” In Immersive Projection Technology / Eurographics Virtual Environments, Zurich, Switzerland, May 2003.
- [PIERCE97] Jeffrey Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik R. and Mine, M.: “Image Plane Interaction Techniques in 3D Immersive Environments”, 1997 Symposium on Interactive 3D Graphics, pages 39-44, 1997.
- [POUP01] Poupyrev, I., Tan, D.S., Billingham M., Kato, H., Regenbrecht, H., Tetsutani, N. “Tiles: A Mixed Reality Authoring Interface”. *Interact 2001*, Tokyo, Japan.
-

- [PREE94] Preece, J., Sharp, H., Benyon, D., Holland, S., Carey, T.: "Human-Computer Interaction". Addison-Wesley; Wokingham, Reading, Menlo Park, New York; 1994
- [RASKAR99] Raskar, R., Welch, G., Chen, W.: "Table-Top Spatially Augmented Reality: Bringing Physical Models to Life with Projected Imagery". Proceedings of Second International IEEE Workshop on Augmented Reality (IWAR'99), San Francisco, pp. 64-71, 1999.
- [REIN98] Reiners, D., Stricker, D., Klinker, G., Müller, S.: "Augmented Reality for Construction Tasks: Doorlock Assembly", 1st International Workshop on Augmented Reality (IWAR'98), San Francisco, 1998.
- [REIN04] Reiners, D.: "Herausforderungen an moderne Szenengraphsysteme am Beispiel OpenSG", Informatik-Spektrum, Volume 27, Issue 6, Dec 2004, Pages 531 - 541
- [REIT01a] G. Reitmayr and D. Schmalstieg, "Mobile collaborative augmented reality," in *Proc. ISAR 2001*, (New York, New York, USA), pp. 114–123, IEEE, October 29–30 2001.
- [REIT01b] Reitmayr, G., Schmalstieg, D.: „OpenTracker – an open software architecture for reconfigurable tracking based on XML”, In *Proc. IEEE Virtual Reality 2001*, pages 285–286, Yokohama, Japan, March 13–17 2001.
- [REIT05] Reitmayr, G., Chiu, C., Kusternig, A., Kusternig, M., Witzmann, H.: „iOrb - Unifying Command and 3D Input for Mobile Augmented Reality“, In *Proc. IEEE VR Workshop on New Directions in 3D User Interfaces* (Bonn, März 2005), pp. 7–10.
- [ROZIER] Rozier, J., Karahalios, K., Donath, J.: „Hear&There: An Augmented Reality System of Linked Audio”
- [SATO98] Sato, Y., Nakamoto, M., Tamaki, Y., Sasama, T., Sakita, I., Nakajima, Y., Monden, M., Tamura, S.: "Image Guidance of Breast Cancer Surgery Using 3-D Ultrasound Images and Augmented Reality Visualization", *IEEE Trans Med Imaging*. 1998 Oktober; 17(5):681-93.
- [SCHMAL00] Schmalstieg, D., Fuhrmann, A., Hesina, G.: "Bridging Multiple User Interfaces With Augmented Reality", In *3rd Int'l Symposium on Augmented Reality*, pp 20-29, Munich, Germany, 2000
- [SCHMAL01] Schmalstieg, D., Billinghurst, M., Poupyrev, I., Kato, H.: „Augmented-Reality: The Interface is everywhere”, *Course Notes of Siggraph 2001*, Los Angeles, USA, August, 2001.
- [SCHNAE02] Schnädelbach, H., Koleva, B., Flintham, M., Fraser, M., Izadi, S., Chandler, P., Foster, M., Benford, S., Greenhalgh, C., Rodden, P.: „The Augurscope: A Mixed Reality Interface for Outdoors”, *Proceedings of CHI02*, Minneapolis, USA, 2002
- [SHNEI83] Shneiderman, B.: „Direct Manipulation: a step beyond programming languages“, *IEEE Computer*, 16(8), pp. 57-69, 1983.
- [SHNEI97] Shneiderman, B.: "Designing the User Interface, 3rd edition", Addison Wesley, 1997

- [SIMON00] Simon, G., Fitzgibbon, A., Zisserman, A.: "Markerless tracking using planar structures in the scene". In *Proc. International Symposium on Augmented Reality*, pages 120–128, October 2000.
- [SLAY01] Slay, H., Thomas, B., Vernik, R.: "Tangible User Interaction Using Augmented Reality", Third Australasian conference on User interfaces, Melbourne, 2002
- [STED99] Stedmon, A., Kalawsky, R., Hill, K., Cook, C.: "Old Theories, New Technologies: Cumulative Clutter Effects Using Augmented Reality," *IEE International Conference on Information Visualization '99*, London, UK, July 1999.
- [STOAK95] Stoakley, R., Conway, M., Pausch, R.: "Virtual reality on a WIM: Interactive worlds in miniature", *Proceedings of CHI'95*, pp. 265-272. ACM SIGCHI, 1995
- [STRICK00] Stricker, D. Fröhlich, T., Söller-Eckert, B.: „The Augmented Man“, 2000
- [STRICK01] Stricker, D., Kettenbach, T.: „Real-Time and Markerless Vision-based Tracking for Out-Door Augmented Reality Applications“, ISAR01, 2001
- [STROU98] Stroustrup, B.: „Die C++ Programmiersprache – 3., aktualisierte und erweiterte Auflage“, Addison-Wesley-Longman, Bonn, 1998
- [SUGA03] Sugano, N., Kato, H., Tachibana, K.: "The Effects of Shadow Representation of Virtual Objects in Augmented Reality", *Proceedings of ISMAR03*, Tokyo, 2003
- [SZAL97] Szalavari, Z., Gervautz, M.: "The Personal Interaction Panel - a Two-Handed Interface for Augmented Reality", *Eurographics 97*, Volume 16, Number 3
- [TANG03] Tang, A., Owen, C., Biotta, F., Mou, W.: "Comparative Effectiveness of Augmented Reality", *SIGCHI03*, 2003
- [TUCE00] Tuceryan, M., Navab, N.: "Single Point Active Alignment Method (SPAAM) for optical see-through HMD Calibration for AR". In *International Symposium for Augmented Reality*, pages 149–158, Munich, 2000.
- [VRML97] The Virtual Reality Modeling Language, International Standard ISO/IEC 14772-1:1997, 1997
- [WAGM03] Wagner, M., Klinker, G.: „An Architecture for Distributed Spatial Configuration of Context Aware Applications“, 2nd International Conference on Mobile and Ubiquitous Multimedia, Norrköping, Sweden, Dec. 11-12 2003
- [WAGD03] Wagner, D., Schmalstieg, D.: "ARToolKit on the PocketPC Platform", *ARToolKit Workshop 2003*, Tokyo
- [WARE88] Ware, C. and Jessome, D.R.: „Using the bat: a six -dimensional mouse for object placement“, *IEEE Computer Graphics and Application*, 11, pages 65-70, 1988.
- [WEIS91] Weiser, M.: „The computer for the 21st century“, *Scientific American*, 265(3):94-104, September 1991.

- [WERN93] Wernecke, J.: "The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor". Addison-Wesley, 2nd edition, November 1993.
- [WLOKA95] Wloka M., Anderson, B.: "Resolving Occlusion in Augmented Reality". Proceedings of Symposium on Interactive 3D Graphics, ACM SIGGRAPH, 1995, pp. 5-12.
- [WOODS03] Woods, E., Mason, P., Billingham, M.: "MagicMouse: an Inexpensive 6-Degree-of-Freedom Mouse", Graphite2003
- [YOU99] You, S., Neumann, U., Azuma, R.: "Hybrid inertial and vision tracking for augmented reality registration". In *Proc.IEEE Virtual Reality '99*, pages 260–267, Houston, TX, March 13–17 1999.
- [ZAUN03] Zauner, J., Haller, M., Brandl, A., Hartmann, W.: "Authoring of a Mixed Reality Assembly Instructor for Hierarchical Structures", Proceedings of ISMAR03, Tokyo, 2003
- [ZHAI96] Zhai, S., Milgram, P., Buxton, W.: „The Influence of Muscle Groups on Performance of Multiple Degree-of-Freedom Input“, Proceedings of SIGCHI'96, 1996
- [ZHAI98] Zhai, S., Milgram, P.: „Quantifying Coordination in Multiple DOF Movement and Its Application to Evaluating 6 DOF Input Devices“, Proceedings of SIGCHI'98, pp. 320-326, 1998
- [FROG] FrogPad, General Information: <http://www.frogpad.com/information/general.asp>
- [GYRATION] GyroRemote, General Information: <http://www.gyration.com/gyroremote.htm>
- [POLH] POLHEMUS Accessoires: <http://www.polhemus.com/?page=Accessories>
- [TRUST] Ami HandTrack Product Overview: <http://www.trust.com/home/>
- [TWID2] Twiddler2 Features: <http://www.handykey.com/site/features.html>
- [WRISTPC] WristPC Keyboard: <http://www.l3sys.com/keybd/keybd.html>

Abbildungsverzeichnis

ABBILDUNG 1: AUGMENTED REALITY BASISFUNKTIONALITÄTEN (KOMPONENTENSICHT)	13
ABBILDUNG 2: PERSONAL INTERACTION PAD (PIP)	20
ABBILDUNG 3: DATENHANDSCHUH.....	21
ABBILDUNG 4: SPACEMOUSE.....	21
ABBILDUNG 5: WRISTPC UNTERARMSTATISTATUR, TWIDDLER2 CHORDIC-KEYBOARD UND FROGPAD	22
ABBILDUNG 6: GYROREMOTE, HANDTRACK	23
ABBILDUNG 7: INTERAKTIONS-GERÄT MIKE	23
ABBILDUNG 8: AUGMENTED REALITY UNTERSTÜTZTES ZUSAMMENSTELLEN VON KABELBÄUMEN [BOEING].....	24
ABBILDUNG 9: PERFORMANZVERGLEICH VERSCHIEDENER ANLEITUNGSFORMEN	24
ABBILDUNG 10: AR UNTERSTÜTZTE WARTUNG EINES LASERDRUCKERS [UNC]	25
ABBILDUNG 11: FUNKTIONALE EINHEITEN DER MENSCHLICHEN INFORMATIONSVERRARBEITUNG UND HANDLUNGSSTEUERUNG [WAND93]	27
ABBILDUNG 12: ANWENDUNGSBEREICHE VON AR	32
ABBILDUNG 13: ROUTINEMÄßIG DURCHGEFÜHRTE FERTIGUNGSTÄTIGKEIT.....	33
ABBILDUNG 14: USE-CASE: SERVICE	37
ABBILDUNG 15: BENUTZERGRUPPEN UND ANWENDUNGSGEBIETE.....	39
ABBILDUNG 16: BEISPIEL-ARBEITSABLAUF	41
ABBILDUNG 17: MODUSWECHSEL.....	41
ABBILDUNG 18: TYPISCHE FORSCHUNGSPROTOTYPEN MOBILER AR SYSTEME (V.L.N.R. TINMITH-EVO5, STUDIERSTUBE, DWARF).....	46
ABBILDUNG 19: PROZESSSICHT	48
ABBILDUNG 20: ARTOOLKIT ARCHITEKTUR	48
ABBILDUNG 21: STUDIERSTUBE SYSTEMARCHITEKTUR [SCHMAL00]	49
ABBILDUNG 22: DWARF MODULE, BESTEHEND AUS SOFT- UND HARDWARE	50
ABBILDUNG 23: TINMITH-EVO5 SYSTEM-ARCHITEKTUR.....	51
ABBILDUNG 24: SERIALISIERUNGSOBJEKTE SORGEN FÜR NETZWERKTRANSPARENZ	51
ABBILDUNG 25: VORGEHENSMODELL ZUR ERSTELLUNG EINES RAHMENSYSTEMS.....	54

ABBILDUNG 26: RAHMENSYSTEM FÜR DIE REALISIERUNG MOBILER AR ANWENDUNGEN	56
ABBILDUNG 27: ZUSTÄNDE EINER KOMPONENTE	58
ABBILDUNG 28: VERBINDUNG VON REALEN ORTEN UND VIRTUELLEN INFORMATIONEN	61
ABBILDUNG 29: RELATIONEN ZWISCHEN REALEN OBJEKTEN UND INFORMATIONEN	63
ABBILDUNG 30: 3D SZENEGEOMETRIE, LINKS VOR, RECHTS NACH DER UMSTRUKTURIERUNG (WEIßE KNOTEN REPRÄSENTIEREN GRUPPENKNOTEN, FARBIGE KNOTEN SIGNALISIEREN DIE ZUGEHÖRIGKEIT ZU UNTERSCHIEDLICHEN OBJEKTEN)	64
ABBILDUNG 31: STRUKTUR EINES INFORMATIONSOBJEKTS	64
ABBILDUNG 32: KONFIGURATIONEN, BEI DENEN DIE SICHTBARKEITSBEDINGUNGEN NICHT ERFÜLLT SIND	67
ABBILDUNG 33: ERFÜLLTE SICHTBARKEITSKRITERIEN UND UNTERSCHIEDLICHE LOD- STUFEN	67
ABBILDUNG 34: XML-BESCHREIBUNG UND KORRESPONDIERENDER DATENFLUSSGRAPH [OTUG03]	70
ABBILDUNG 35: NETZWERKWEITE GERÄTEANBINDUNG ÜBER IDEAL	71
ABBILDUNG 36: DIREKTE GERÄTEANBINDUNG ÜBER IDEAL-LLDI	71
ABBILDUNG 37: ÜBERGÄNGE ZWISCHEN DEN VERSCHIEDENEN PHASEN DES VISIONBASIERTEN TRACKINGS	73
ABBILDUNG 38: ABLAUF DES KOMBINIERTEN TRACKING-VERFAHRENS OHNE BESCHLEUNIGUNGSSENSOR	74
ABBILDUNG 39: DATENFLÜSSE IM TRACKING-FRAMEWORK	75
ABBILDUNG 40: DATENAUSTAUSCH ÜBER DEN GEMEINSAMEN DATEN-POOL	76
ABBILDUNG 41: SYNCHRONISIERUNG ÜBER DATENOBJEKTE	77
ABBILDUNG 42: DIREKTER ZUGRIFF AUF ANDERE KOMPONENTE	78
ABBILDUNG 43: VERSUCHSAUFBAU MIT ZWEI KAMERAS UND INERTIAL-SENSOR	79
ABBILDUNG 44: VERDECKUNGSDARSTELLUNG ÜBER PHANTOMGEOMETRIE	80
ABBILDUNG 45: VERDECKUNGEN UND VOLUMENSCHATTEN	80
ABBILDUNG 46: STANDARDMÄßIGE KONFIGURATION DER DARSTELLUNGS-LAYER	82
ABBILDUNG 47: SZENENGRAPHSTRUKTUR	82
ABBILDUNG 48: STRUKTUR DES SZENENGRAPHEN MIT VERDECKUNGSDARSTELLUNG	83
ABBILDUNG 49: TRAVERSIERUNG DER TEILBÄUME FÜR VERDECKUNGS- (LINKS) UND SZENENDARSTELLUNG (RECHTS)	84
ABBILDUNG 50: GERÄTEANBINDUNG AUF SZENENGRAPHEBENE	85
ABBILDUNG 51: VERBINDUNG VON TEILSZENENGRAPHEN UND TRACKING-SENSOREN	86
ABBILDUNG 52: STRUKTUR DER TEILBÄUME	86
ABBILDUNG 53: AR-VIEWER STELLT ALLE UNTERSTÜTZTEN DOKUMENTTYPEN DAR	87

ABBILDUNG 54: INTEGRATION VON DOKUMENTTYPEN ÜBER WRAPPER UND ENTSPRECHENDE SZENENGRAPH-KNOTEN	88
ABBILDUNG 55: ARVIKA-ARCHITEKTUR ZUR UNTERSTÜTZUNG VON MOBILEN SYSTEMEN	89
ABBILDUNG 56: AR-BROWSER INTEGRATION IN HTML-INHALTE	90
ABBILDUNG 57: FLYSTICK 6DOF INTERAKTIONSGERÄT UND DESSEN ECHO IN VR ANWENDUNG	96
ABBILDUNG 58: KOMFORTABLE ARBEITSBEREICHE (ROT) UND BLICKFELD (GRÜN)	98
ABBILDUNG 59: VISIONBASIERTES TRACKING DES ARBEITSBEREICHS ÜBER ZUSÄTZLICHE KAMERA (GELB).	98
ABBILDUNG 60: HYBRIDE USER-INTERFACES REALISIERT ÜBER PDA UND GRAFIKTABLETT. 100	
ABBILDUNG 61: SCHEMATISCHER SYSTEMAUFBAU FÜR DIE REALISIERUNG EINES HYBRIDEN USERINTERFACE MIT EINEM ZWEITEN RECHNER.....	100
ABBILDUNG 62: IORB INTERAKTIONSGERÄT [REIT05].....	103
ABBILDUNG 63: IORB FREIHEITSGRADE.....	103
ABBILDUNG 64: VERSCHIEDENE MARKTVERFÜGBARE INERTIAL-TRACKINGSYSTEME [INTERSENSE, XSSENS].....	104
ABBILDUNG 65: TRANSFORMATION ZWISCHEN TRACKING-SYSTEM UND INTERAKTIONSGERÄT	106
ABBILDUNG 66: STEUERUNG DER SELEKTION ÜBER ANSTEIGENDE FLANKEN DER TASTE.....	107
ABBILDUNG 67: STEUERUNG DER SELEKTION ÜBER STEIGENDE UND FALLENDE FLANKE DES TASTENIMPULSES	108
ABBILDUNG 68: INTERPRETATION VERSCHIEDENER AUSLENKUNGEN IM RELATIVEN MODUS	108
ABBILDUNG 69: MAPPING VON WINKEL AUF GESCHWINDIGKEIT	109
ABBILDUNG 70: ALTERNATIVE MENU-DARSTELLUNGEN FÜR UNTERSCHIEDLICHE ROTATIONSACHSEN	110
ABBILDUNG 71: HEAD-STABILIZED GUI IN VERBINDUNG MIT TABLETT	111
ABBILDUNG 72: GENERIERUNG VERSCHIEDENER DARSTELLUNGEN AUS EINER ABSTRAKTEN BESCHREIBUNG	113
ABBILDUNG 73: USERINTERFACE GENERIERUNG AUS ABSTRAKTER BESCHREIBUNG	114
ABBILDUNG 74: UNTERSTÜTZUNG VERSCHIEDENER GUI BESCHREIBUNGSSPRACHEN.....	114
ABBILDUNG 75: QT DESIGNER [TROLLTECH]	115
ABBILDUNG 76: LAUFZEITSYSTEM ZUR ERZEUGUNG VON 3D GUIS	115
ABBILDUNG 77: STRUKTUR EINES GUI-ELEMENTS.....	116
ABBILDUNG 78: QT GUI UND ENTSPRECHENDE 3D REPRÄSENTATION.....	117
ABBILDUNG 79: BEZIEHUNG ZWISCHEN PROTOTYPEN, GEOMETRIEN UND STYLES	118

ABBILDUNG 80: ZUSAMMENWIRKEN ZWISCHEN LAYOUT-MANAGER UND GUI-ELEMENT	118
ABBILDUNG 81: 3D GUI IM AR-BROWSER	119
ABBILDUNG 82: ABLAUF DER INSTANZIIERUNG EINES GUI-ELEMENTS.....	119
ABBILDUNG 83: HIERARCHISCHER AUFBAU EINER SERVICE-TÄTIGKEIT	125
ABBILDUNG 84: EXEMPLARISCHER AUFBAU EINES TEMPLATES	126
ABBILDUNG 85: DARSTELLUNG IN AUTORENTOOL UND LAUFZEITSYSTEM	128
ABBILDUNG 86: ERZEUGEN DER ARBEITSANWEISUNGEN ALS VRML-BESCHREIBUNG.....	129
ABBILDUNG 87: ERZEUGEN DER ARBEITSANWEISUNG ALS ABSTRAKTE XML-BESCHREIBUNG	129

Eigene Veröffentlichungen

- Makri, A., Arsenijevic, D., Weidenhausen, J., Eschler, P., Stricker, D., Machui, O., Fernandes, C., Maria, S., Voss, G., Ioannidis, N.: „ULTRA: An Augmented Reality System for Handheld Platforms, Targeting Industrial Maintenance Applications“, 11th International Conference on Virtual Systems and Multimedia, Ghent, 2005
- Makri, A., Weidenhausen, J., Eschler, P., Stricker, D., Machui, O., Fernandes, C., Maria, S., Voss, G., Ioannidis, N.: “ULTRA Light Augmented Reality Mobile System”, 4th IEEE and ACM International Symposium on Mixed and Augmented Reality, Wien, 2005.
- Becker, M., Bleser, G., Pagani, A., Pastarmov, Y., Stricker, D., Vial, F., Weidenhausen, J., Wohlleber, C., Wuest, H.: “Visual Tracking for Augmented Reality: No Universal Solution but Many Powerful Building Blocks”, 2. Workshop der GI-Fachgruppe VR/AR.Aachen: Shaker, 2005, pp. 107-118
- Behr, J., Dähne, P., Kresse, W., Stricker, D., Weidenhausen, J., Zieringer, A.: “Mixed Reality: Ein Überblick über die aktuellen Anwendungen und Trends”, Symposium Mensch & Fahrzeug, Darmstadt, 2005
- Knöpfle, C., Weidenhausen, J., Chauvigné, L., Stock, I.: “Template Based Authoring for AR based Service Scenarios”, Proceedings of IEEE Virtual Reality 2005, Los Alamitos, 2005, pp. 237-240
- Lutz, B., Weidenhausen, J., Müller, P., Gora, S., Vereenoghe, T., Stricker, D., Van Gool, L.: “EPOCH Showcase: On Site Experience”, 5th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage - VAST2004, Brüssel, 2004
- Stricker, D., Hamadou, M., Hoymann, H., Jahn, D., Kiesel, B., Krieg, S., Lescher, M., Niemann, M., Plewinski, N., Schubert, M., Vilsmeier, C., Weidenhausen, J., Wichert, R., „Augmented Reality Basistechnologien“, In: Wolfgang, Friedrich (Hrsg.): ARVIKA - Augmented Reality für Entwicklung, Produktion und Service. Erlangen, München : Publicis Verlag, 2004, S. 52-93
- Weidenhausen, J.: “Enhancing Tangible User Interfaces with Physically Based Modeling”. HCI International 2003. Proceedings of the 10th International Conference on Human-Computer Interaction. Volume 2, pp. 801-805
- Weidenhausen, J., Knöpfle, C., Stricker, D.: “Lessons learned on the way to industrial augmented reality applications, a retrospective on ARVIKA”, Computers & Graphics 27 (2003) pp. 887-891

- Hamadou, M., Jahn, D., Weidenhausen, J.: "ARVIKA - Augmented Reality für Entwicklung, Produktion und Service", i-com, (2002), 2, S. 4-10
- Haase, H., Bock, M., Hergenröther, E., Knöpfle, C., Koppert, H.-J., Schröder, F., Trembilski, A., Weidenhausen, J.: "Meteorology meets computer graphics - a look at a wide range of weather visualisations for diverse audiences". Computers & Graphics 24 (2000), pp. 391-397.
- Haase, H., Bock, M., Hergenröther, E., Knöpfle, C., Koppert, H.-J., Schröder, F., Trembilski, A., Weidenhausen, J.: "Where Weather Meets the Eye -- A Case Study on a Wide Range of Meteorological Visualisations for Diverse Audiences". Proceedings of IEEE/EG VisSym'99 (Springer-Verlag), Wien

Lebenslauf

Persönliche Daten

Name: Jens-Martin Weidenhausen
Geburtsdatum: 23. Oktober 1969
Geburtsort: Frankfurt / Main
Familienstand: Verheiratet, ein Kind

Schulausbildung

1976 - 1990 Grundschohle Frankfurt / Main
1980 - 1989 Gymnasium Frankfurt / Main

Berufsausbildung und Studium

1991 - 1998 Studium der Informatik an der TU Darmstadt mit Abschluss als Diplom Informatiker

Berufstätigkeit

1999 - 2001 Wissenschaftlicher Mitarbeiter im Zentrum für Graphische
Datenverarbeitung e.V. in Darmstadt
2001 – 2006 Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Graphische
Datenverarbeitung in Darmstadt